Spring 2008

Numerical and adaptive grid methods for ideal magnetohydrodynamics

Burlen Loring
University of New Hampshire, Durham

NUMERICAL AND ADAPTIVE GRID METHODS FOR IDEAL MAGNETOHYDRODYNAMICS

BY

BURLEN LORING

B.S., University of New Hampshire (2004)

THESIS

Submitted to the University of New Hampshire
in partial fulfillment of
the requirements for the degree of

Master of Science

in

Mathematics

May 2008

This thesis has been examined and approved.

Thesis Director, Joachim Raeder

Associate Professor of Physics

Marianna Shubov

Professor of Mathematics

Eric Grinberg

Professor of Mathematics

03/24/08

Date

DEDICATION

To those who have helped me along the way, especially Ed Kelley, Jim Harbison, Janice Loring and Nellie Crossley. And, to Katherine Singer for being very supportive throughout this process.

ACKNOWLEDGEMENTS

I would like to thank: Professor Joachim Raeder for advice, support, and access to his MHD code, and in addition, the ideas for the new numerical methods which have been developed in this thesis; Dan Martin of LBNL for advanced access to CHOMBO edge data classes; Manuel Torrilhon of Seminar for Applied Mathematics, ETH Zurich for giving me access to unpublished exact solutions to the Brio and Wu MHD Riemann problem; John Dorrelli, Yann Rasera, and Professor Steve Wineberg for encouragement and many useful discussion on MHD numerics.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

NUMERICAL AND ADAPTIVE GRID METHODS FOR IDEAL MAGNETOHYDRODYNAMICS

by

Burlen Loring
University of New Hampshire, May, 2008

In this thesis numerical finite difference methods for ideal magnetohydrodynamics(MHD) are investigated. A review of the relevant physics, essential for interpreting the results of numerical solutions and constructing validation cases, is presented. This review includes a discusion of the propagation of small amplitude waves in the MHD system as well as a thorough discussion of MHD shocks, contacts and rarefactions and how they can be pieced to gether to obtain a solutions to the MHD Riemann problem. Numerical issues relevant to the MHD system such as: the loss of nonlinear numerical stability in the presence of discontinuous solutions, the introduction of spurious forces due to the growth of the divergence of the magnetic flux density, the loss of pressure positivity, and the effects of non-conservative numerical methods are discussed, along with the practical approaches which can be used to remedy or minimize the negative consequences of each. The use of block structured adaptive mesh refinement is investigated in the context of a divergence free MHD code. A new method for conserving magnetic flux across AMR grid interfaces is developed and a detailed discussion of our implementation of this method using the CHOMBO AMR framework is given. A preliminary validation of the new method for conserving magnetic flux density across AMR grid interfaces illustrates that the method works. Finally a number of code validation cases are examined spurring a discussion of the strengths and weaknesses of the numerics employed.

CHAPTER 1

IDEAL MAGNETOHYDRODYNAMICS

1.1 Chapter Abstract

In this chapter the Ideal MHD system is introduced, a review of the physics which describe the behavior of such systems is presented. I will introduce various system formulations which although are technically equivalent facilitate the investigation of disparate aspects of the system's behavior in very individual ways. The conservation form is emphasized for its importance in numerical simulations where shocks are concerned. I will discuss the physical and mathematical aspects of the MHD system which set it apart from its cousin the Ideal Hydrodynamic system. I will apply current theory in the area of analysis of non-linear multidimensional hyperbolic systems to the Ideal MHD system in its full three dimensional form, illustrating the inherent multidimensionality of the system and facilitating a discussion of the physics. To that end, I will discuss "magneto-acoustic" wave propagation, recent development on the theory of MHD shocks, and the implications of the divergence free constraint. This chapter represents the knowledge that is requisite when reading current literature and developing numerical solutions of the ideal MHD system. In chapter 5 the foundation established here will be built upon when presenting validation runs for code I have written.

1.2 Ideal MHD and Its Formulation In Terms of Primitive Fields

Magnetohydrodynamics is the study of the macroscopic interaction between an electrically conducting liquid or gas and a magnetic field. Ideal MHD describes these interactions in

the non-diffusive limit, or more specifically, when the fluid is an inviscid polytropic ideal gas with equation of state,

$$\frac{p}{\rho} = (c_p - c_v)T, \tag{1.1}$$

the flow speeds are non-relativistic, variation in the electric and magnetic field strength occur on the same time and length scales as variations in the gas, changes in the gas state occur adiabatically, and the conductance of the gas is infinite (Jeffrey, 1966). Ohm's law is, in this case, given by

$$\mathbf{E} = \mathbf{B} \times \mathbf{v}. \tag{1.2}$$

The ideal MHD approximation is valid when the scale of temporal evolutions of system components are large compared to the characteristic interaction time of individual gas molecules and spatial scales are large compared characteristic lengths of interaction of an individual gas molecules[1]. Given these restrictions ideal MHD is expressed by the following system of equations, known as the Lundquist equations[2]:

$$\frac{D\rho}{Dt} + \rho\nabla \cdot \mathbf{v} = 0 \tag{1.3}$$

$$\frac{D\mathbf{v}}{Dt} - \frac{1}{\rho}(\nabla \times \mathbf{B}) \times \mathbf{B} + \frac{1}{\rho}\nabla p(\rho, S) = 0 \tag{1.4}$$

$$\rho T\frac{DS}{Dt} = \rho\frac{De}{Dt} - \frac{p}{\rho}\frac{D\rho}{Dt} = 0 \tag{1.5}$$

$$\frac{\partial \mathbf{B}}{\partial t} - \nabla \times (\mathbf{v} \times \mathbf{B}) = 0 \tag{1.6}$$

$$\nabla \cdot \mathbf{B} = 0 \tag{1.7}$$

where $\frac{D}{Dt} = \frac{\partial}{\partial t} + (\mathbf{v} \cdot \nabla)$ gives the rate of change of a quantity enclosed by a fluid element in motion. The so called primitive fields are: the density of the gas, ρ, the bulk gas velocity, $\mathbf{v}$, the gas pressure, p, the entropy per unit mass, S, and the magnetic field strength or

[1]While this is a rather crude statement, I stop here because my focus is on "how to" rather than "when to" use ideal MHD. The interested reader can find a detailed discussion of the derivation, assumptions, and applicability of ideal MHD in Boyd and Sanderson (2003, chap. 3).

[2]Here and in the sections to follow the equations have been normalized.

magnetic flux density[3], $\mathbf{B}$. The remaining variables are the derived quantities: the specific energy density, e, and the temperature of the gas, T. The constants specific to a given gas are, c_p, the specific heat at constant pressure, c_v, the specific heat at constant volume, and their ratio, γ, the adiabatic exponent. The right hand side of (1.5) is identically zero, however I have left the terms involving ρ and e as a reminder as to how (1.5) relates to the equation for conservation of energy. Additionally the relation between entropy and energy is completed by:

$$U = \frac{1}{2}\rho\mathbf{v}\cdot\mathbf{v} + \frac{1}{2}\mathbf{B}\cdot\mathbf{B} + \rho e, \tag{1.8}$$

where,

$$e = c_v T = \frac{p}{\rho(\gamma - 1)} \tag{1.9}$$

and U is the total energy per unit volume, the first term on the right side of equation (1.8) is the kinetic energy of the fluid, the second term is the energy density of the magnetic field, and the third term is the internal energy density of the fluid. Equation (1.7) is a constraint on equation (1.6), the nature of which will be discussed in detail in Section 1.6. As will be illustrated in Sections 1.7-1.9, it is the primitive variable formulation of the ideal MHD system which is convenient for an analytic treatment fleshing out many interesting and useful properties regarding the propagation of small amplitude waves through the system.

1.3 Formulation In Terms of Conservative Fields

Systems of conservation laws are the mathematical expression of dynamic systems where the properties of the system are transported or moved about without the possibility of creation or destruction. In general if one can say that some quantity u_i is conserved then what is meant is that given some fixed arbitrary volume over which u_i quantified then at some later time if the amount of u_i has changed, it was due to flow through the surface of

[3]It is common when discussing the magnetic flux density to say simply, "magnetic field", however one must be careful to not confuse the magnetic flux density with the magnetic field lines which are purely imaginary, and introduced for our mental convenience.

the prescribed volume. For a vector of conserved quantities,

$$\mathbf{u}(\mathbf{x},t) \equiv (u_1(\mathbf{x},t), u_2(\mathbf{x},t), ..., u_p(\mathbf{x},t))^T, \tag{1.10}$$

with flux functions given by,

$$\mathbf{f}\left(\mathbf{u}(\mathbf{x},t)\right) \equiv (\mathbf{f}_1(\mathbf{u}(\mathbf{x},t)), \mathbf{f}_2(\mathbf{u}(\mathbf{x},t)), ..., \mathbf{f}_p(\mathbf{u}(\mathbf{x},t)))^T, \tag{1.11}$$

we have:

$$\iiint_V \mathbf{u}(\mathbf{x}, t_0+dt) - \mathbf{u}(\mathbf{x},t_0)\, dV + \int_{t_0}^{t_0+dt} \iint_S \mathbf{f}\left(\mathbf{u}(\mathbf{x},t)\right) \cdot dS\, dt = 0 \tag{1.12}$$

where the first integral quantifies the fields $\mathbf{u}$ at the start and end of a time interval on the volume V, and the second integral quantifies the flow or flux across the surface S bounding V during that time interval. By applying the fundamental Theorem of Calculus to the first term and the Divergence Theorem to the second, and rearranging we arrive at

$$\int_{t_0}^{t_0+dt} \iiint_V \left[\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \mathbf{f}\left(\mathbf{u}\right)\right] dV\, dt = 0 \tag{1.13}$$

Here the volume is chosen is arbitrarily, as a result the equation holds for any such volume, and, the integrand is identically zero. We finally arrive at the commonly used differential form for conservation laws, namely,

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \mathbf{f}\left(\mathbf{u}\right) = 0. \tag{1.14}$$

The equations of Ideal MHD arranged in conservation form are:

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \mathbf{f}\left(\mathbf{u}\right) = 0 \tag{1.15}$$

$$\nabla \cdot \mathbf{B} = 0 \tag{1.16}$$

where:

$$\mathbf{u} = \begin{bmatrix} \rho \\ \rho\mathbf{v} \\ \mathbf{B} \\ U \end{bmatrix}, \quad \mathbf{f}\left(u\right) = \begin{bmatrix} \rho\mathbf{v} \\ \rho\mathbf{v}\mathbf{v} + I(p + \frac{B^2}{2}) - \mathbf{B}\mathbf{B}) \\ \mathbf{v}\mathbf{B} - \mathbf{B}\mathbf{v} \\ \left(U + p + \frac{B^2}{2}\right)\mathbf{v} - \mathbf{B}(\mathbf{v}\cdot\mathbf{B}) \end{bmatrix}, \tag{1.17}$$

and,

$$p = (\gamma - 1)\left[U - \frac{1}{2}\rho\mathbf{v}\cdot\mathbf{v} - \frac{1}{2}\mathbf{B}\cdot\mathbf{B}\right]. \tag{1.18}$$

Here the conservative fields are, ρ, the fluid density, $\rho\mathbf{v}$, the momentum per unit volume, U, the total energy per unit volume, and $\mathbf{B}$, the magnetic flux density. The remaining variable, p, the gas pressure, is a derived quantity. Immediately one can see that this system is not exactly captured by the differential conservation law form of equation (1.14), due to the constraint equation (1.16). We'll take a closer look at the constraint equation in Section 1.6.

The conservation form is, in a sense, the closest to the underlying physics of any of the formulations, as the MHD system has been derived directly from the principles of conservation of mass, momentum, energy and magnetic flux density. Additionally, the conservation formulation is the key to analyzing the system's behavior in the presence of shocks and other discontinuities which, although physically valid, leave the differential equations undefined. Further, numerical finite difference methods not in conservation form, do not converge to the true solution, giving false shock locations and incorrect jump strengths, where such discontinuities are involved. See LeVeque (1992, p. 35) for an example.

1.4 A Semi-Conservative Formulation

The momentum equation in conservation form is,

$$\frac{\partial\rho\mathbf{v}}{\partial t} + \nabla\cdot\left[\rho\mathbf{v}\mathbf{v} + I\left(p + \frac{B^2}{2}\right) - \mathbf{B}\mathbf{B}\right] = 0. \tag{1.19}$$

Applying the divergence to the terms involving $\mathbf{B}$ we obtain,

$$\nabla\cdot\left[I(\frac{B^2}{2}) - \mathbf{B}\mathbf{B}\right] = - \begin{bmatrix} B_x(\nabla\cdot\mathbf{B}) - B_y(\frac{\partial B_y}{\partial x} - \frac{\partial B_x}{\partial y}) + B_z(\frac{\partial B_x}{\partial z} - \frac{\partial B_z}{\partial x}) \\ B_x(\frac{\partial B_y}{\partial x} - \frac{\partial B_x}{\partial y}) + B_y(\nabla\cdot\mathbf{B}) - B_z(\frac{\partial B_z}{\partial y} - \frac{\partial B_y}{\partial z}) \\ -B_x(\frac{\partial B_x}{\partial z} - \frac{\partial B_z}{\partial x}) + B_y(\frac{\partial B_z}{\partial y} - \frac{\partial B_y}{\partial z}) + B_z(\nabla\cdot\mathbf{B}) \end{bmatrix}. \tag{1.20}$$

When the constraint equation is satisfied, this expression is proportional to the forces exerted on the gas by the magnetic field. According to the constraint equation, (1.16), the terms containing the divergence of the magnetic flux density vanish. Dropping them, one can write:

$$\vec{\mathbf{F}}_{\mathbf{B}} = -\nabla \cdot \left[I(\frac{B^2}{2}) - \mathbf{BB} \right] = (\nabla \times \mathbf{B}) \times \mathbf{B} = \mathbf{j} \times \mathbf{B}. \tag{1.21}$$

Use of this expression in equation (1.19) leads directly to the semi-conservative formulation of the momentum equation,

$$\frac{\partial \rho \mathbf{v}}{\partial t} + \nabla \cdot [\rho \mathbf{vv} + Ip] - \mathbf{j} \times \mathbf{B} = 0. \tag{1.22}$$

Here, the terminology "semi-conservative" comes about due to the fact that, where numerical solution is concerned, momentum is conserved to within the local truncation error of the numerical scheme, $O(\Delta x^m, \Delta t^n)$ (Brackbill and Barnes, 1989). Also, if the magnetic flux density is zero, then equation (1.22) is reduced to the hydrodynamic momentum equation in conservative form. Note, that in equation (1.20) the divergence of the magnetic flux density appears in each of the components. If the constraint equation is not satisfied then these terms contribute a spurious force proportional to the magnitude of the divergence directed along the magnetic field. If large enough this non-physical force can exert undue influence on the system with potentially disastrous results (Brackbill and Barnes, 1989).

The energy equation in conservation form is,

$$\frac{\partial U}{\partial t} + \nabla \cdot \left[\left(U + p + \frac{B^2}{2} \right) \mathbf{v} - \mathbf{B}(\mathbf{v} \cdot \mathbf{B}) \right] = 0. \tag{1.23}$$

Applying the divergence to the terms involving $\mathbf{B}$ we obtain an expression which describes the work done on the gas by the magnetic field, namely,

$$\begin{aligned} W_{\mathbf{B}} &= \nabla \cdot \left[\mathbf{v}B^2 - \mathbf{B}(\mathbf{v} \cdot \mathbf{B}) \right] \\ &= B^2(\nabla \cdot \mathbf{v}) + \mathbf{v} \cdot (\nabla B^2) - (\mathbf{v} \cdot \mathbf{B})(\nabla \cdot \mathbf{B}) - \mathbf{B} \cdot \nabla(\mathbf{v} \cdot \mathbf{B}). \end{aligned} \tag{1.24}$$

Note that the third term is proportional to the divergence of the magnetic flux density. This term accounts for the work done on the gas by the spurious force discussed above. In

light of the constraint equation, (1.16), this term vanishes. Using equation (1.6) and some identities from vector calculus we can write,

$$\nabla \cdot [(\mathbf{B} \cdot \mathbf{B})\mathbf{v} - \mathbf{B}(\mathbf{v} \cdot \mathbf{B})] = \frac{\partial}{\partial t}\left(\frac{B^2}{2}\right) - \mathbf{j} \cdot \mathbf{E}. \tag{1.25}$$

The use of this expression in equation (1.23) leads directly to the semi-conservative formulation of the energy equation,

$$\frac{\partial e_T}{\partial t} + \nabla \cdot [e_T + p]\,\mathbf{v} - \mathbf{j} \cdot \mathbf{E} = 0. \tag{1.26}$$

where,

$$e_T = \frac{\rho v^2}{2} + \frac{p}{\gamma - 1}. \tag{1.27}$$

Using equations (1.22), (1.26), and (1.27), the equations of ideal MHD can be written as

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \mathbf{f}(\mathbf{u}) + \mathbf{A} = 0 \tag{1.28}$$

$$\frac{\partial \mathbf{B}}{\partial t} + \nabla \times \mathbf{E} = 0 \tag{1.29}$$

$$\nabla \cdot \mathbf{B} = 0, \tag{1.30}$$

where

$$\mathbf{u} = \begin{bmatrix} \rho \\ \rho\mathbf{v} \\ e_T \end{bmatrix} \quad \mathbf{f}(\mathbf{u}) = \begin{bmatrix} \rho\mathbf{v} \\ \rho\mathbf{v}\mathbf{v} + Ip \\ (e_T + p)\mathbf{v} \end{bmatrix} \quad \mathbf{A} = \begin{bmatrix} 0 \\ -\mathbf{j} \times \mathbf{B} \\ -\mathbf{j} \cdot \mathbf{E} \end{bmatrix} \tag{1.31}$$

$$p = (\gamma - 1)\left[e_T - \frac{1}{2}\rho\mathbf{v} \cdot \mathbf{v}\right], \quad e_T = \frac{\rho v^2}{2} + \frac{p}{\gamma - 1}, \quad \mathbf{j} = \nabla \times \mathbf{B} \tag{1.32}$$

and, ρ, is the fluid density, $\rho\mathbf{v}$, the momentum per unit volume, and $\mathbf{B}$, the magnetic flux density. The remaining variables are the derived quantities: p, the gas pressure, e_T, the sum of gas internal and kinetic energy per unit volume, and $\mathbf{j}$, the current density. Written this way, mass and magnetic flux are conserved. However, where numerical solution is concerned, momentum and energy are only conserved to within the local truncation error of the particular fintie difference scheme used. It is this form of the ideal MHD equations which has been implemented in our code.

The semi-conservative formulation of the equation of Ideal MHD has the advantage that if the divergence of the magnetic flux density is not zero, as is often the case where numerical solution is concerned, some harmful effects on the solution are eliminated. Namely, those due to the introduction of a spurious force which is directed along the magnetic field, as illustrated in equation (1.20), and of the exchange of energy with that force as illustrated in equation (1.25), the presence of which can lead to a serious or catastrophic deterioration in the accuracy of the solution (Brackbill and Barnes, 1989), (Toth, 2000). Note, that in addition to the changes to the momentum and energy equation, with reference to the conservative formulation, the formula for calculating the pressure has also changed. This endows pressure positivity to the formulation when the magnetic energy makes up a large fraction of the total energy. The numerical issue of pressure positivity is discused further in Section 2.8. As mentioned above, deviations from the conservation form have negative consequences where shocks are involved. However, in this case the loss of conservation is bound by the local truncation error of the numerical methods employed, and therefore the use of the semi-conservative formulation will give reasonable results in many cases where shocks are concerned.

1.5 The 8-Wave Formulation

To account for the negative effects that a non-zero divergence has on the numerical solution of the equations of ideal MHD, the structure of the equations can be modified such that an extraneous eigenvalue is introduced while leaving the ideal MHD eigenvalues unchanged[4]. These changes to the underlying system of equations have the effect of freezing the divergence into the gas so that it is advected with the flow, much in the same manner that entropy waves are advected with the flow. This results in a loss of conservation and a new, albeit, very similar system of equations. The advantage of this formulation is realized only for high velocity flow problems with open flow type boundary conditions. In that case, the

[4]The eigenvalues of the ideal MHD system are derived in section 1.8

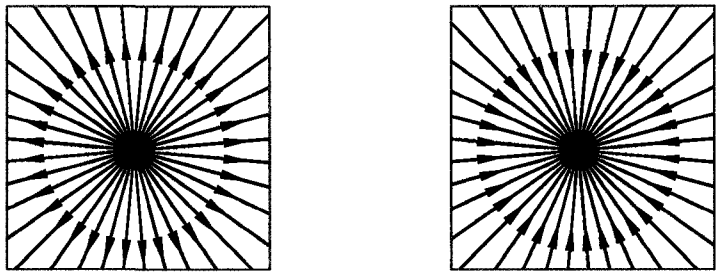


Figure 1-1: Two magnetic field line topologies impossible to obtain while satisfying the divergence free constraint.

divergence, rather than accumulating in place and destroying the accuracy of the solution as it grows, will be advected with the flow passing out of the system through the open boundaries. Variants of this formulation have been constructed such that as divergence of the magnetic flux density is advected it is also diffused or damped out. More information can be found in Powell (1994), Dellar (2001) and Toth (2000).

1.6 The Divergence Free Constraint

Of all of the fields involved in the various formulations of the equations of ideal MHD, the magnetic flux density is unique. It, alone, is subject to the divergence free constraint of equation (1.34). In light of this, there are really two equations responsible for the evolution of the magnetic flux density namely,

$$\frac{\partial \mathbf{B}}{\partial t} + \nabla \times \mathbf{E} = 0, \tag{1.33}$$

and,

$$\nabla \cdot \mathbf{B} = 0. \tag{1.34}$$

The constraint equation, (1.34), is a direct expression of the fact that, unlike electric fields, there is no particle that can be said to be the source of the magnetic flux density, i.e. there are no magnetic monopoles. A topological consequence of this limitation is that magnetic field line configurations such as those depicted in Figure 1-1 do not occur, and magnetic field lines are always loops which close upon them selves (Davidson, 2001). The

lack of magnetic monopoles can be expressed in terms of the magnetic flux by the equation,

$$\Phi = \iint_s \mathbf{B} \cdot dS = \iiint_V [\nabla \cdot \mathbf{B}]\, dV = 0. \tag{1.35}$$

Where V is some finite volume and S the surface enclosing it. If the divergence of the magnetic flux density is non-zero on a finite volume then there must be a magnetic monopole inside. Equation (1.35) is useful when considering numerical methods because it brings a finite volume into the mix on which the divergence must be zero, presenting a practical way for numerical methods to couple the constraint equation to the evolution of the magnetic flux density. When constructing numerical methods, satisfying the divergence free constraint on a finite volume is key to the methods success. For instance see the discussion in Toth (2000). This will be clearly illustrated in Section 2.5.

Often it is said that equation (1.34) is an initial condition, such that if it is true intially it will be so at any later time, because taking the divergence of Faraday's law we have:

$$\nabla \cdot \frac{\partial \mathbf{B}}{\partial t} = -\nabla \cdot \nabla \times \mathbf{E}, \tag{1.36}$$

from which

$$\frac{\partial}{\partial t} \nabla \cdot \mathbf{B} = 0. \tag{1.37}$$

When solving the MHD equations directly by analytic integration this is certainly true and there is no cause for concern. However, when solving the MHD equations numerically this postulate fails us, for numerical finite difference methods have the property that the time rate of change of the divergence of the magnetic flux density on some finite volume is not identically zero, but rather

$$\frac{\partial}{\partial t} \nabla \cdot \mathbf{B} = \xi(t), \tag{1.38}$$

where ξ is some non-zero function of time. Regardless of the exact form of ξ, the divergence introduced in a single finite difference step is bound in between the accumulated rounding errors during the computation of the curl of the electric field, and the local truncation error(LTE) of the finite difference scheme employed. In either case the divergence of the magnetic flux density grows as a function of the number of finite difference steps taken.

That is to say that, once introduced these errors are accumulated. Depending on the specific implementation this growth can lead to a serious deterioration in the accuracy of the solution due to the effects of the associated spurious forces as illustrated in equation (1.20) (Brackbill and Barnes, 1989). Therefor, where numerical solution is concerned, it is not sufficient to take the view that (1.34) is solely an initial condition. One must take care that the constraint given by equation (1.34) is satisfied adequately over the entire time interval through which the solution is eveloved.

Many methods have been proposed to deal with this issue. For instance: the 8-wave formulation and its variants (Powell, 1994), where divergence in the magnetic flux density is advected out of the system; the so called Hodge projection (Brackbill and Barnes, 1989), where a potential equation is solved from which a new 'cleaned' magnetic field is periodically constructed; and constrained transport, where Stokes's law and Divergence theorem are used to construct a numerical method which limits the growth in divergence on a finite volume to the order of rounding errors (Evans and Hawley, 1988). From a practical point of view the 8-wave approach is the simplest to implement, the Hodge projection is the most computationally costly, and the constrained transport method adds the most complexity. When solving the ideal MHD system numerically one of these approaches is usually be employed. The constrained transport method has been used in our code, the details of which are discussed in Section 2.5.

1.7 Hyperbolic Systems

In general a system of p partial differential equations of the form:

$$\frac{\partial \mathbf{u}}{\partial t} + \sum_{j=1}^{d} \frac{\partial \mathbf{f}_j(\mathbf{u})}{\partial x_j} = \mathbf{0}, \tag{1.39}$$

where $\mathbf{u} = (u_1, ..., u_p)^T$ and $\mathbf{f}_j = (f_{1j}(\mathbf{u}), ..., f_{pj}(\mathbf{u}))^T$, can be re-formulated into what is known as the quasi-linear form, namely,

$$\frac{\partial \mathbf{u}}{\partial t} + \sum_{j=1}^{d} \mathbf{f}_j'(\mathbf{u}) \frac{\partial \mathbf{u}}{\partial x_j} = \mathbf{0}, \tag{1.40}$$

by using the chain rule to write:

$$\frac{\partial \mathbf{f_j(u)}}{\partial x_j} = \mathbf{f}_j'(\mathbf{u})\frac{\partial \mathbf{u}}{\partial x_j}, \tag{1.41}$$

where

$$\mathbf{f}_j'(\mathbf{u}) = \frac{\partial \mathbf{f}_j(\mathbf{u})}{\partial \mathbf{u}} = \begin{pmatrix} \partial/\partial u_1 \\ \partial/\partial u_2 \\ \vdots \\ \partial/\partial u_p \end{pmatrix} (f_{1j}(\mathbf{u}), f_{2j}(\mathbf{u}), ..., f_{pj}(\mathbf{u})) \,. \tag{1.42}$$

From this the multi-dimensional Jacobian matrix is defined as (Godlewski, 1996):

$$\underline{\mathbf{A}}(\mathbf{u}, \omega) = \sum_{j=1}^{d} \mathbf{f}_j'(\mathbf{u})\omega_j, \tag{1.43}$$

where, $\omega \equiv (\hat{e}_1, \hat{e}_2, \hat{e}_3)$, are the basis vectors of the chosen Cartesian coordinate system. An analysis of the Jacobian matrix reveals much about the properties and solutions of a given system of equations. For the MHD system a multidimensional treatment is particularly relevant.

A system of partial differential equations of the form (1.39) is said to be hyperbolic when for any $\mathbf{u}$ and any direction ω, the Jacobian matrix $\underline{\mathbf{A}}(u, \omega)$ has p real eigenvalues $\lambda_1(\mathbf{u}, \omega), ..., \lambda_p(\mathbf{u}, \omega)$ with a complete set of right eigenvectors $\mathbf{r}_i(\mathbf{u}, \omega)$ where

$$\underline{\mathbf{A}}(\mathbf{u}, \omega)\mathbf{r}_k(\mathbf{u}, \omega) = \lambda_k(\mathbf{u}, \omega)\mathbf{r}_k(\mathbf{u}, \omega) \tag{1.44}$$

A system of the form (1.39) is said to be strictly-hyperbolic if the eigenvalues are distinct for all $\mathbf{u}$ (Godlewski, 1996).

For a strictly hyperbolic system we can write the system in terms of characteristic variables ϕ and t where $\phi(t) = (\xi(t), \eta(t), \zeta(t))$ is some parametrized curve such that[5] $u_i(\phi(t), t) = const.$ The hypersurface defined by $\mathbf{u}(\phi(t), t) = const.$ evaluated at a specific instant of time will be called a characteristic wavefront. We have the following property on the hypersurface and by extension each wavefront:

$$\frac{d\mathbf{u}}{dt} = \sum_{j=1}^{d} \frac{\partial \mathbf{u}}{\partial \phi_j}\frac{\partial \phi_j}{\partial t} + \frac{\partial \mathbf{u}}{\partial t} = 0. \tag{1.45}$$

[5]Note that in one dimension $\phi(t) = x(t)$.

From which we have that,

$$\frac{\partial \mathbf{u}}{\partial t} = -\sum_{j=1}^{d} \frac{\partial \mathbf{u}}{\partial \phi_j} \frac{\partial \phi_j}{\partial t}. \tag{1.46}$$

Using the chain rule, we can also write,

$$\frac{\partial \mathbf{u}}{\partial x_j} = \sum_{i=1}^{d} \frac{\partial \mathbf{u}}{\partial \phi_i} \frac{\partial \phi_i}{\partial x_j}. \tag{1.47}$$

Substitution of these expressions into (1.40) leads to:

$$\sum_{i=1}^{d} \left\{ -I \frac{\partial \phi_i}{\partial t} + \sum_{j=1}^{d} \mathbf{f}_j'(\mathbf{u}) \frac{\partial \phi_i}{\partial x_j} \right\} \frac{\partial \mathbf{u}}{\partial \phi_i} = 0 \tag{1.48}$$

Using the above definition of the Jacobian we let $\vec{\nu} = (\frac{\partial \phi}{\partial x_1}, ..., \frac{\partial \phi}{\partial x_j})$ and write:

$$\left(\underline{\mathbf{A}}(\mathbf{u}, \vec{\nu}) - I \frac{\partial \phi_i}{\partial t} \right) \frac{\partial \mathbf{u}}{\partial \phi_i} = 0. \tag{1.49}$$

Hence, $\partial/\partial t(\phi \cdot \omega) = \lambda(\mathbf{u}, \vec{\omega})$ is an eigenvalue of $A(\mathbf{u}, \nu)$. The specific choice of $\vec{\nu}$ coincides, by definition, with the gradient of the level surface defined by the characteristic wavefront, and as such is normal to it. Thus, $\lambda_i(\mathbf{u}, \vec{\omega})$ is the rate of change of the characteristic wavefront in the direction normal to it. It is because of this relationship that eigenvalues are called characteristic velocities.

1.8 Eigenvalues of the MHD System In Three Dimensions

For the ideal MHD the multi-dimensional Jacobian is given by:

$$\underline{\mathbf{A}}(\mathbf{u}, \omega) = \begin{pmatrix} \mathbf{v} \cdot \vec{\omega} & \rho\omega_1 & \rho\omega_2 & \\ \frac{a^2 \omega_1}{\rho} & \mathbf{v} \cdot \vec{\omega} & 0 & \\ \frac{a^2 \omega_2}{\rho} & 0 & \mathbf{v} \cdot \vec{\omega} & \\ \frac{a^2 \omega_3}{\rho} & 0 & 0 & \\ 0 & 0 & 0 & \\ 0 & B_y\omega_1 - \mathbf{B} \cdot \vec{\omega} & B_x\omega_2 & \\ 0 & B_y\omega_1 & B_y\omega_2 - \mathbf{B} \cdot \vec{\omega} & \\ 0 & B_z\omega_1 & B_y\omega_2 & \end{pmatrix} \quad ... \tag{1.50}$$

$$\begin{pmatrix}
\rho\omega_3 & 0 & 0 & 0 & 0 \\
0 & \frac{\delta p\omega_1}{\rho} & \frac{B_x\omega_1}{\rho} - \frac{\mathbf{B}\cdot\vec{\omega}}{\rho} & \frac{B_y\omega_1}{\rho} & \frac{B_z\omega_1}{\rho} \\
0 & \frac{\delta p\omega_2}{\rho} & \frac{B_x\omega_2}{\rho} & \frac{B_y\omega_2}{\rho} - \frac{\mathbf{B}\cdot\vec{\omega}}{\rho} & \frac{B_x\omega_2}{\rho} \\
\mathbf{v}\cdot\vec{\omega} & \frac{\delta p\omega_3}{\rho} & \frac{B_x\omega_3}{\rho} & \frac{B_y\omega_3}{\rho} & \frac{B_z\omega_3}{\rho} - \frac{\mathbf{B}\cdot\vec{\omega}}{\rho} \\
0 & \mathbf{v}\cdot\vec{\omega} & 0 & 0 & 0 \\
B_x\omega_3 & 0 & \mathbf{v}\cdot\vec{\omega} - v_x\omega_1 & -v_x\omega_2 & -v_x\omega_3 \\
B_y\omega_3 & 0 & -v_y\omega_1 & \mathbf{v}\cdot\vec{\omega} - v_y\omega_2 & -v_y\omega_3 \\
B_z\omega_3 - \mathbf{B}\cdot\vec{\omega} & 0 & -v_z\omega_1 & -v_z\omega_2 & \mathbf{v}\cdot\vec{\omega} - v_z\omega_3
\end{pmatrix}$$

after algebraic manipulation and, with the help of Maple, the eigenvalues are found to be:

$$\lambda_1 = 0 \tag{1.51}$$

$$\lambda_2 = \mathbf{v}\cdot\omega \tag{1.52}$$

$$\lambda_{3,4} = \mathbf{v}\cdot\omega \pm \left\{\frac{1}{2}\left(\frac{\mathbf{B}\cdot\mathbf{B}}{\rho} + a^2 - \left\{\left(\frac{\mathbf{B}\cdot\mathbf{B}}{\rho} + a^2\right)^2 - \frac{4a^2(\omega\cdot\mathbf{B})^2}{\rho}\right\}^{1/2}\right)\right\}^{1/2} \tag{1.53}$$

$$\lambda_{5,6} = \mathbf{v}\cdot\omega \pm \sqrt{\frac{(\omega\cdot\mathbf{B})^2}{\rho}} \tag{1.54}$$

$$\lambda_{7,8} = \mathbf{v}\cdot\omega \pm \left\{\frac{1}{2}\left(\frac{\mathbf{B}\cdot\mathbf{B}}{\rho} + a^2 + \left\{\left(\frac{\mathbf{B}\cdot\mathbf{B}}{\rho} + a^2\right)^2 - \frac{4a^2(\omega\cdot\mathbf{B})^2}{\rho}\right\}^{1/2}\right)\right\}^{1/2} \tag{1.55}$$

λ_1 is zero and does not contribute a characteristic wave to the solution. λ_2 leads to wavefronts which are simply advected with the flow speed $\mathbf{v}$. For the remainder of the eigenvalues it can be seen that the characteristic speeds are highly anisotropic. For instance

$$\omega\cdot\mathbf{B} = \sqrt{\mathbf{B}\cdot\mathbf{B}}cos(\theta) \tag{1.56}$$

so that these characteristic velocities are functions of the angle θ between direction of propagation, ω, and the magnetic flux density, $\mathbf{B}$. Setting $\mathbf{v} = 0$ in (1.53), (1.54) and (1.55) one obtains the so called slow, Alfven and fast speeds. This terminology has come about due to the relation

$$\lambda_8 \leq \lambda_6 \leq \lambda_4 \leq \lambda_2 \leq \lambda_3 \leq \lambda_5 \leq \lambda_7. \tag{1.57}$$

Note that since in ϕ,t space information is propagated on characteristic wavefronts at finite velocities given by the characteristic velocities speeds, the state of a given point in ϕ,t space can only have been influenced by points lying inside a region enclosed by the fastest characteristic surface traced backward in time from the point in question (Laney, 1998, p. 28). This region is called the domain of dependence. In light of the symmetry of the characteristics around the flow speed given by (1.57), the domain of dependence is the region enclosed by characteristic surfaces associated with fast eigenvalues, λ_8 and λ_7. This region can be explicitly found by tracing these characteristic surfaces backward in time. For instance, see Figure 2-2.b.

1.9 Characteristic Wavefronts

We can now apply the special relationship that the eigenvalues have to the characteristic wavefronts namely that velocity normal to the wavefront is given by $\lambda(\mathbf{u}, \omega)$ where ω is taken to be in the direction normal to the wavefront. We will consider an initial state with $\mathbf{u}$ constant everywhere with $\mathbf{v} = 0$ and perturb a sphere of radius R_0 located at the origin of some coordinate system by incerasing the pressure there so that inside this sphere $p = p + \delta p$. This pressure disturbance will launch characteristic waves into the surrounding constant state. When the pressure perturbation is small the fields of $\mathbf{u}$ all remain smooth during the evolution but will posses discontinuous first or higher derivatives (Jeffrey and Taniuti, 1964). The resulting traveling waves are known as weak discontinuities, here we will determine the spatial location of this discontinuity which is cooincident with the wavefront of "magneto-acoustic" wavefronts.

The wavefront of the initial pressure disturbance is given by the surface of the spherical shell enclosing the refion inside of which the pressure was perturbed. The initial normal direction, ω, to this wavefront at each point on the circumference is in the radial direction. With the wavefront surface and its normal vectors we can directly apply (1.52)-(1.55) to determine the evolution of the wavefront through the constant state for each of the characteristic fields. Because of the anisotropy the evolved wavefront is not spherical.

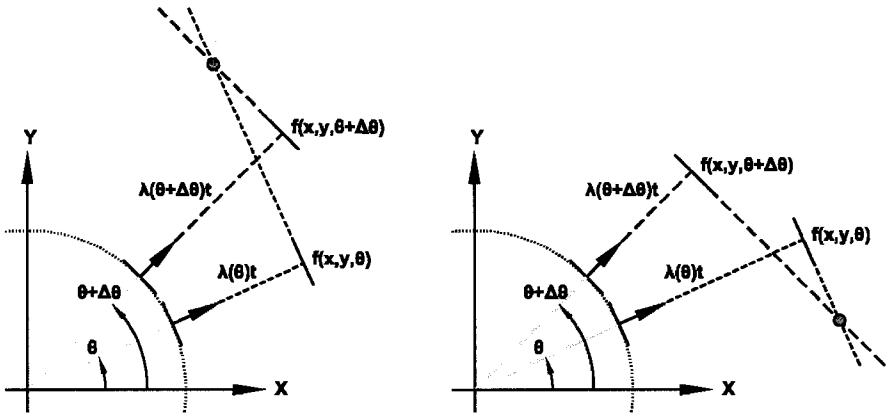


Figure 1-2: Two near by plane wave approximations to an evolved characteristic surface after elapsed time t in directions θ and $\theta + \Delta\theta$. On the left, $\lambda(\theta) < \lambda(\theta + \Delta\theta)$. On the right $\lambda(\theta) > \lambda(\theta + \Delta\theta)$.

Following in the steps of (Jeffrey, 1966) we will determine the location of the weak disturbance after a period of time has elapsed. This will give some critical insight into the nature of wave propagation in the MHD system that proves useful in validating MHD codes. We will consider the evolution of a plane wave emanating from each point on the initial wave front which due to the symmetry in the initial condition has been reduced to a circle with radius R_0. After some time t has elapsed the plane wave has traveled a distance of $\lambda(\theta)t$ in the radial direction. The plane wave wavefront is initially given by the plane tangent to the circle and the eveloved plane wavefront at time t is given by:

$$xcos(\theta) + ysin(\theta) - (R_0 + \lambda(\theta)t) = 0 \tag{1.58}$$

The evolved characteristic wavefront will be given by the envelope of all the evolved plane wave fronts taken from each point on the initial characteristic wavefront, given by the circle $r = R_0$ at time $t = 0$. In (1.58) we have a plane which is parametrized by the variable θ which we treat as the function $f(x, y, \theta) = 0$. The situation is depicted in Figure 1-2. Two cases shown, in the first the characteristic velocity, $\lambda(\theta)$, increases over a small change in direction, $\Delta\theta$. An approximation to a point lying on the evolved characteristic wavefront is located at the intersection of the two nearby plane wavefronts, shown in green. The second case shows the situation where the characteristic velocity decreases over a small change in

direction. From these figures we see how the shape of the evolved characteristic wavefront, given by the envelope of the evolved plane wavefronts, differs substantially from the shape of the initial characteristic wavefront when the characteristic velocities are anisotropic.

We can use a Taylor series expansion to find that a near by evolved planar wavefront is given by:

$$f(x,y,\theta+\Delta\theta) = f(x,y,\theta) + \frac{\partial}{\partial\theta}f(x,y,\theta)\Delta\theta + \frac{1}{2}\frac{\partial^2}{\partial\theta^2}f(\xi(\theta+\Delta\theta))(\Delta\theta)^2 \tag{1.59}$$

The envelope of the plane wave wavefronts is the locus of points given by the limit $\Delta\theta \rightarrow 0$ of:

$$f(x,y,\theta) = f(x,y,\theta+\Delta\theta) = 0. \tag{1.60}$$

Substituting (1.60) into (1.59) we have:

$$\frac{\partial}{\partial\theta}f(x,y,\theta) = 0 \tag{1.61}$$

or in terms of our specific plane waves defined by (1.58) we have:

$$-xsin(\theta) + ycos(\theta) - \frac{\partial\lambda}{\partial\theta}t = 0. \tag{1.62}$$

Now we have two equations namely (1.58) and (1.62), which together we can solve for x and y to find:

$$x = R_0cos(\theta) + \left(\lambda(\theta)cos(\theta) - \frac{\partial\lambda}{\partial\theta}sin(\theta)\right)t \tag{1.63}$$

$$y = R_0sin(\theta) + \left(\lambda(\theta)sin(\theta) + \frac{\partial\lambda}{\partial\theta}cos(\theta)\right)t, \tag{1.64}$$

These equations describe the shape and location of the characteristic surfaces after an elapsed time t.

Using (1.53), (1.54) and (1.55) in these expression provides the solutions which are shown Figure 1-3 along with plots of the wavefront normal velocity, $\lambda(\theta)$ for three different cases. From top to bottom $B_x^2/\rho = a^2$, $B_x^2/\rho > a^2$ and $B_x^2/\rho < a^2$. In all of the plots R_0 is taken to be zero. The shape of the evolved wavefronts are substantially different from

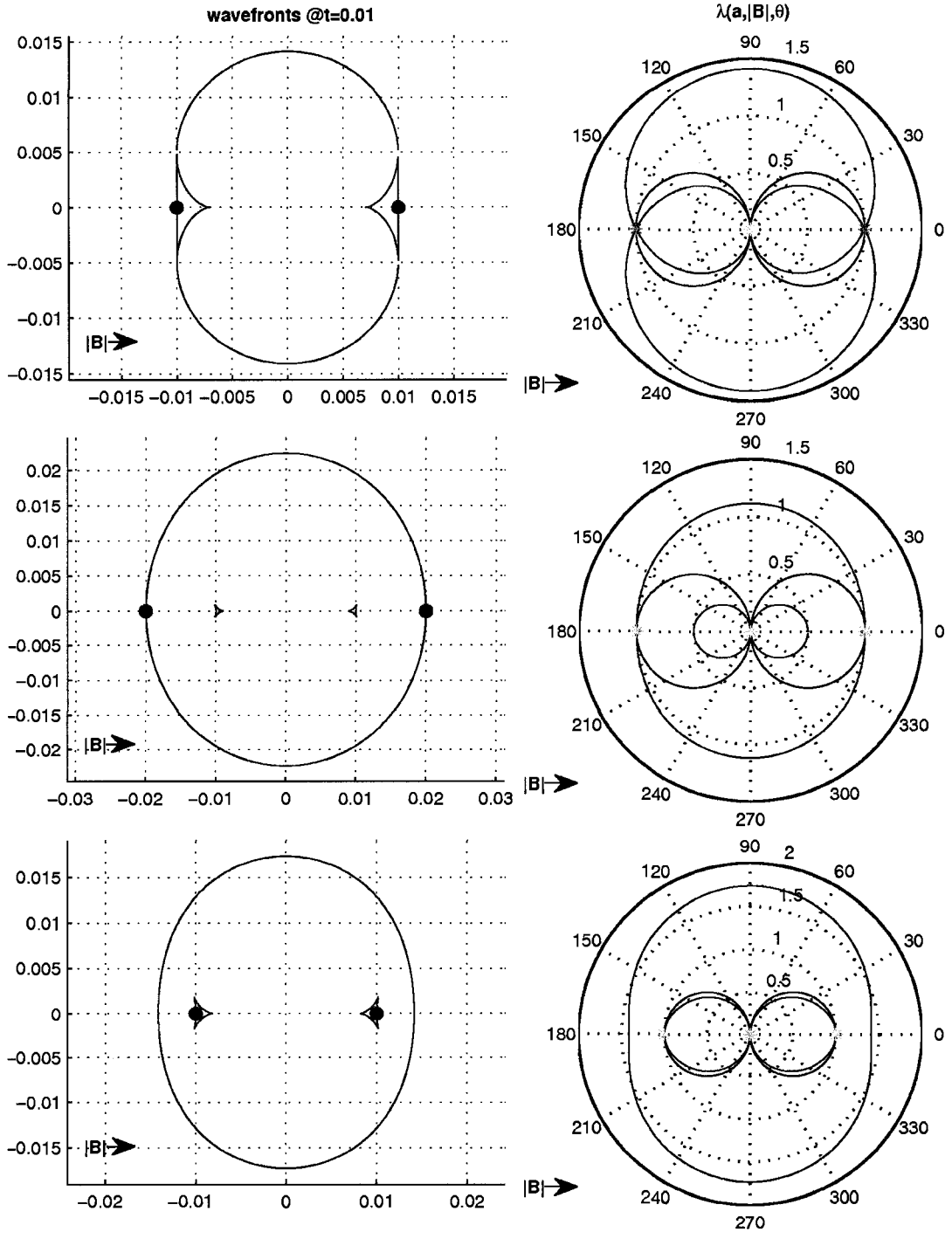


Figure 1-3: Characteristic wavefronts (left column) and surfaces of normal velocity (right column). First row $\mathbf{B}.\mathbf{B}^2/\rho = a^2$; Second row $\mathbf{B}.\mathbf{B}/\rho > a^2$; Third row $\mathbf{B}.\mathbf{B}/\rho < a^2$. In all figures slow mode is red, alfven mode is black, and fast mode is blue. The black dots are the point disturbances of the Alfven modes. The green asterisk marks a triple umbilic and the green asterisks mark double umbilics.

their initial spherical shape. For for the wavefront associated with the fast mode, (1.55), the evolved wavefront is oblong moving fastest perpendicular to the magnetic field. For the wavefront associated with the slow mode, (1.53), two tri-cusped fronts develop and move in both parallel and antiparallel directions with respect to the magnetic field. For the case of the wavefront associated with the Alfven mode, (1.54), the initially spherical wavefront devolves into two points moving in a direction parallel to the magnetic field. Note that in the velocity plots there are certain directions for which not all of eigenvalues are distinct. For instance in all of the plots shown in figure there are values of θ which lead to either $\lambda_3 = \lambda_5$ or $\lambda_5 = \lambda_7$ or $\lambda_3 = \lambda_5 = \lambda_7$. These are known as umbilic points. The presence of the umbilic points complicates a theoretical analysis of the system because at these points the system's eigenvalues are not distinct and therefore system fails to be strictly hyperbolic while much of the mathematical theory applies only to strictly hyperbolic systems. Further at the triple umbilic the system does not have a complete set of eigenvectors (Brio and Wu, 1988). This loss of hyperbolicity is one of the most interesting and consequential features of the MHD system. One consequence is the introduction of the possibility of a number of unconventional shocks not found in strictly hyperbolic systems. The implications of the complications introduced by the loss of strict hyperbolicity will be discussed in Section 1.11.

The characteristic wavefronts give us a multi-dimensional solution which can be used as a piece of a code validations suite, allowing us to verify that small amplitude MHD waves are modeled correctly. However, it is only a partial solution in that from equations (1.63) and (1.64), we have only the wavefront locations and not their amplitudes. A more complete solution to the weak mode disturbance problem is given in Bazer and Fleischamn (1959) which includes the amplitudes of resulting waves.

1.10 The Riemann Problem, Discontinuous Solutions

For systems of linear equations a transformation into characteristic variables completely decouples the equations leading to a simplified system where characteristic surfaces of the same field never intersect. For nonlinear hyperbolic systems the equations are not com-

pletely decoupled as after the transform the characteristic velocities, λ_i are still functions of characteristic fields,$\mathbf{u}$. Therefor it is possible for characteristic surfaces of the same field to interact with each other in $\mathbf{x}$-t space in complicated ways. When one characteristic surface overtakes another a strong discontinuity or jump in one or more of characteristic fields known as a shock will form. To be precise, a shock is a type of strong discontinuity where in a frame of reference moving with the discontinuity there is mass flow across surface of the discontinuity (Jeffrey, 1966, p. 202). A strong discontinuity where there is no mass flow across the surface of the discontinuity is called a contact (Jeffrey, 1966, p. 217). The process by which one characteristic surface overtakes another to form a shock is known as wavefront steepening. Figure 1-5 cases S and F, for instance show the slow and fast characteristic fields(thin lines) converging into shocks(thick line). Although derivatives of $\mathbf{u}$ do not exist across strong discontinuities the system is still well defined there, a result which can be proved using the integral form of the equation. Solutions containing strong discontinuities are called weak solutions (Jeffrey and Taniuti, 1964, p. 119). The presence of strong discontinuities can be problematic for numerical finite difference methods since numerical errors can be unbound near such discontinuities. This will be discussed in Section 2.4.

While weak solutions can form from smooth initial conditions they may also be excited in the system via a discontinuous initial condition because, not all discontinuities are stable (Jeffrey, 1966, p. 219). An instable discontinuity will break up into a series of stable discontinuities which satisfy the principle of conservation and the second law of thermodynamics. When a system such as (1.39) is paired with a discontinuous initial condition given by two states a left, $\mathbf{u}^L$, and a right, $\mathbf{u}^R$, separated by a plane across which for at least one i, $u_i^L \neq u_i^R$, so that at least one of the fields experiences an instantaneous jump crossing the plane of interface, the resulting problem is known as the Riemann problem. Generally, the discontinuity specified in the initial condition of a specific Riemann problem is not stable and will break up into a series of states separated by propagating stable discontinuities(shocks, contacts) and/or propagating smooth transitions(rarefactions). Figure

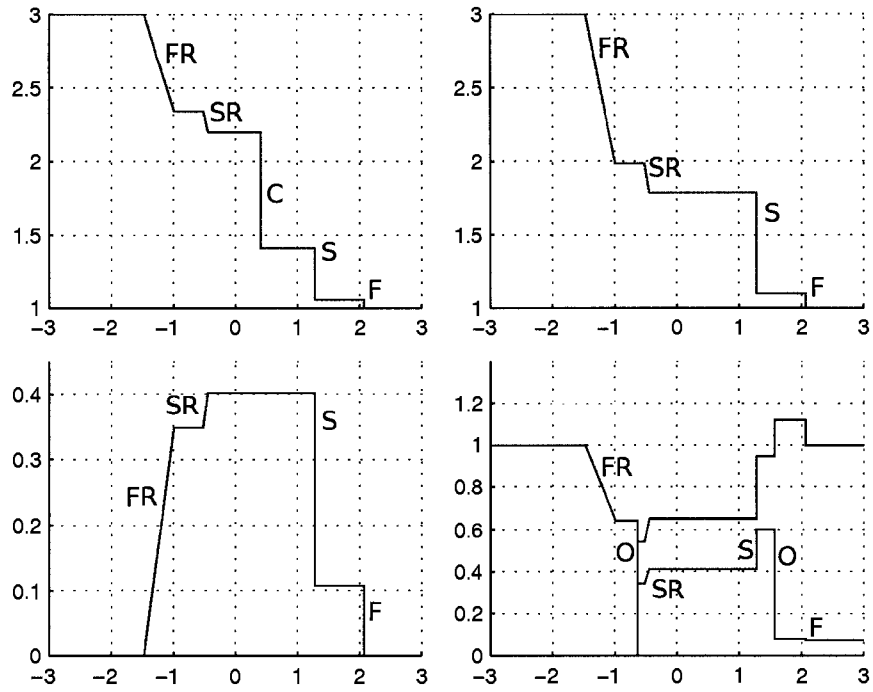


Figure 1-4: Exact solution to an MHD Riemann Problem at $t = 1$ (Torrilhon, 2002). Upper left ρ, upper right p, lower left v_x, lower right B_y red, B_z black. Discontinuities are labeled, F,S fast and slow shock, C contact, FR,SR fast and slow rarefactions, O over compressive shock. Compare with Figure 1-5.

1-4 shows the solution of a specific Riemann problem (Torrilhon, 2002) for 5 of the eight fields of the ideal MHD system. The discontinuities in the region $x < 0$ are traveling to the left while the discontinuities in the region $x > 0$ are traveling to the right.

The study of a general Riemann problem allows one to investigate the behavior of weak solutions to systems of the form of equation (1.39). The analytic solution of specific MHD Riemann problems gives one a method to validate shock capturing numerical codes. The eigen-structure of the ideal MHD system gives rise to many unconventional discontinuous features in the solution of the Riemann problem. Paired with the fact that many numerical methods break down near discontinuities this leads one to the conclusion that a fair understanding of the Riemann problem is required when validating numerical finite difference methods. Without such an understanding it is difficult to discern which effects in a particular numerical solution are expressions of the underlying physics and which are simply deficiencies of a given finite difference method, or coding error. While a complete analysis

of the MHD Riemann problem is beyond the scope of this thesis, some effort has been made to introduce and discuss various aspects that arise in the context of developing shock capturing numerics so that the solutions may be used as a piece of a code validation suite.

1.11 Shocks and Contact Discontinuities

For strictly hyperbolic systems with p fields the solution to the Riemann problem may contain as many as $p+2$ states of which some subset will be separated by strong discontinuities. Any discontinuity realized in the solution must be such that the conservation principle is satisfied on an arbitrary volume enclosing the discontinuity, leading to specific restrictions of flow across the discontinuity. Contacts satisfy conservation principles trivially since there is no mass flow across the discontinuity. An explicit expression for the mass flow across a shock is obtained by integrating the conservation form of a general hyperbolic equation across a hypothetical discontinuity bound by an arbitrary volume and applying the divergence theorem, (Jeffrey, 1966, p. 197). Specifically conservation across the shock requires that the following condition be satisfied,

$$(u_i^L - u_i^R)\mathbf{s}\cdot\mathbf{n} = \left(\mathbf{F}_i(\mathbf{u}^L) - \mathbf{F}_i(\mathbf{u}^R)\right)\cdot\mathbf{n}, \tag{1.65}$$

where $\mathbf{n}$ is the direction normal to the shock and,

$$\mathbf{F}_i(\mathbf{u}) \equiv \sum_{j=1}^{3} f_{ij}(\mathbf{u})\hat{\mathbf{e}}_j, \tag{1.66}$$

where $\hat{\mathbf{e}}_j$ are basis vectors. This is known as the Rankine-Hugoniot condition and it relates the propagation speed of the discontinuity, $\mathbf{s}$, to the jump in the fields, and the flow across (normal to) the shock surface. If $\mathbf{u}^L$ is known then one can use the Rankine-Hugoniot relations to form the set of all $\mathbf{u}^R$ that satisfy the conservation property across a shock[6]. The resulting $\mathbf{u}^R$ form a collection of points in phase space which are known as a Hugoniot curve. Two adjacent states in the solution to the Riemann problem separated by a shock

[6]In the following discussion only right running waves are considered since the Riemann problem can be solved in two halves, and the process for either is essentially the same (Myong and Roe, 1997a).

must lie on the same Hugoniot curve[7]. However, there are multiple Hugoniot curves through each point $\mathbf{u}$ in phase space. Thus an additional criterion must be used to decide which Hugoniot will yield a physically admissible solution.

For strictly hyperbolic systems this additional criterion is known as the Lax entropy condition. The Lax entropy condition was developed in gas dynamics as an expression of the second law of thermodynamics which states that entropy in a system can never decrease. Despite this connection to gasdynamics it can be proved that for any hyperbolic system sufficient conditions to guarantee that the application of the Lax entropy condition result in uniquely determined solutions to the Riemann problem, are that the system of equations in question be strictly hyperbolic with genuinely nonlinear or linear degenerate characteristic fields (Godlewski, 1996, p. 84). A characteristic field, λ_i is said to be genuinely nonlinear if

$$\nabla_{\mathbf{u}}\lambda_i(\mathbf{u},\omega)\cdot\mathbf{r}_i(\mathbf{u},\omega)\neq 0,\forall\mathbf{u} \tag{1.67}$$

while characteristic field is said to be linearly degenerate if

$$\nabla_{\mathbf{u}}\lambda_i(\mathbf{u},\omega)\cdot\mathbf{r}_i(\mathbf{u},\omega)=0,\forall\mathbf{u}. \tag{1.68}$$

Genuine nonlinearity implies that the characteristic velocities vary monotonically while linear degeneracy implies that characteristic velocities are constant (LeVeque, 1992, p. 76). If all the characteristic fields of a given system of equations are either genuinely nonlinear or linearly degenerate then the system's flux function:

$$\mathbf{f}(\mathbf{u},\omega)=\sum_{j=1}^{3}\mathbf{f}_j(\mathbf{u})\omega_j \tag{1.69}$$

is said to be convex. If the conditions of strict hyperbolicity of the system and convexity of the flux function are satisfied then the Lax entropy condition can be expressed as:

$$\lambda_i(\mathbf{u}^L)>\mathbf{s}\cdot\mathbf{n}>\lambda_i(\mathbf{u}^R) \tag{1.70}$$

which implies that for a given shock characteristic surfaces of the same family converge into the discontinuity (LeVeque, 1992). However, as shown above the eigenstructure of the ideal

[7]For some specific examples of the Hugoniot curves from ideal MHD see Figure 1 of (Myong and Roe, 1997b).

MHD equations is such that the system is not everywhere strictly hyperbolic due to the presence of umbilic points. At triple umbilic points, the strict hyperbolicity of the system fails and the characteristic fields associated with the fast and slow magnetosonic speeds are not genuinely nonlinear since there, the system fails to have a complete set of right eigenvectors (Brio and Wu, 1988). In light of this the Lax entropy condition is not sufficient to identify all of the relevant solutions to the Riemann problem for ideal MHD. As a result extensions to the Lax condition were developed using the second law of thermodynamics and shock stability arguments. These are known as evolutionary admissibility conditions (Jeffrey and Taniuti, 1964). However it has since been shown that the evolutionary admissibility conditions also end up being incomplete, leaving a large number of physically relevant Riemann problems unsolvable (Jeffrey and Taniuti, 1964), (Myong and Roe, 1997a), (Brio and Wu, 1988). Specifically the evolutionary condition discards all of the so called intermediate type shocks which are discussed below. These are physically valid shocks and thus are needed to solve certain Riemann problems of interest.

One approach for determining shock admissibility in the ideal MHD system is to examine shock behavior in the diffusive MHD system, since shocks which are not realizable in the diffusive MHD system should also not be realizable in the ideal MHD system, and shocks which are realizable in the full MHD system should, in their non-diffusive limit, be realizable the ideal MHD system. To this end, one makes an analysis of traveling wavefronts in the MHD system including diffusive physics namely, viscosity, resistivity and thermal conductivity. The diffusive terms are (Jeffrey, 1966): for the momentum equation,

$$-\eta\nabla^2\mathbf{v} - \frac{\eta}{3}\nabla(\nabla\cdot\mathbf{v}), \tag{1.71}$$

for the energy equation,

$$\nabla\cdot\left(-\eta\sum_{j,k=1}^{3}\left(\frac{\partial v_k}{\partial x_j}+\frac{\partial v_j}{\partial x_k}-\frac{2}{3}\delta_{j,k}\nabla\cdot\mathbf{v}\right)v_k\hat{\mathbf{e}}_j\right), \tag{1.72}$$

and

$$-\kappa\nabla^2 T \tag{1.73}$$

and for the magnetic flux density equation,

$$-\frac{1}{\sigma}\nabla^2\mathbf{B}, \tag{1.74}$$

where $\hat{\mathbf{e}}_j$ are Cartesian basis vectors, T is the gas temperature and η, κ and $1/\sigma$ are respectively the coefficients of viscosity, of thermal conductivity and of electrical resistivity for a given gas[8]. In many real world situations the parameters η, κ, $1/\sigma$ are very small. Thus for smooth solutions these terms do not contribute much to the solution. However, as wavefront steeping occurs and a shock begins to form, the factors of ∇^2 cause these terms become significantly large, tending to keep the solution smooth (LeVeque, 1992, p. 26), (Witham, 1974, p. 26). The presence of the diffusive terms give shocks a smooth transition with a thickness on the order of a few mean free path lengths of the particular gas in question (Jeffrey, 1966, p. 195). This thin transition region is known as the viscous profile of the shock. The analysis of traveling waves of the diffusive MHD system is carried out by formulating a dynamical system from the diffusive MHD equations an then applying the Rankine-Hugoniot conditions in a frame moving with the shock, analyzing the singularities of this dynamical system, and finally discarding solutions which violate the second law of thermodynamics (Myong and Roe, 1997a). The remaining traveling waves are considered admissible under the viscosity admissibility condition. These can all be characterized by the fact that across the shock at least one family of characteristics must converge and across the shock entropy must increase (Myong and Roe, 1997a).

The shocks which are admissible under the viscosity admissibility condition are shown in Figure 1-5. The first two columns of the first row correspond to the Slow and Fast shocks Lax type shocks. These two cases satisfy the Lax entropy condition and can be distinctly associated with a single characteristic field, illustrating that the viscosity admissibility condition encompasses the Lax condition. It has been shown that magnetic flux density increases across a fast Lax type shock and decreases across a slow Lax type shock. Further, in both cases the jump in the magnetic flux density occurs solely in the tangential

[8]See (Jeffrey, 1966) sec 7 and 8 for more information and a description of the notation

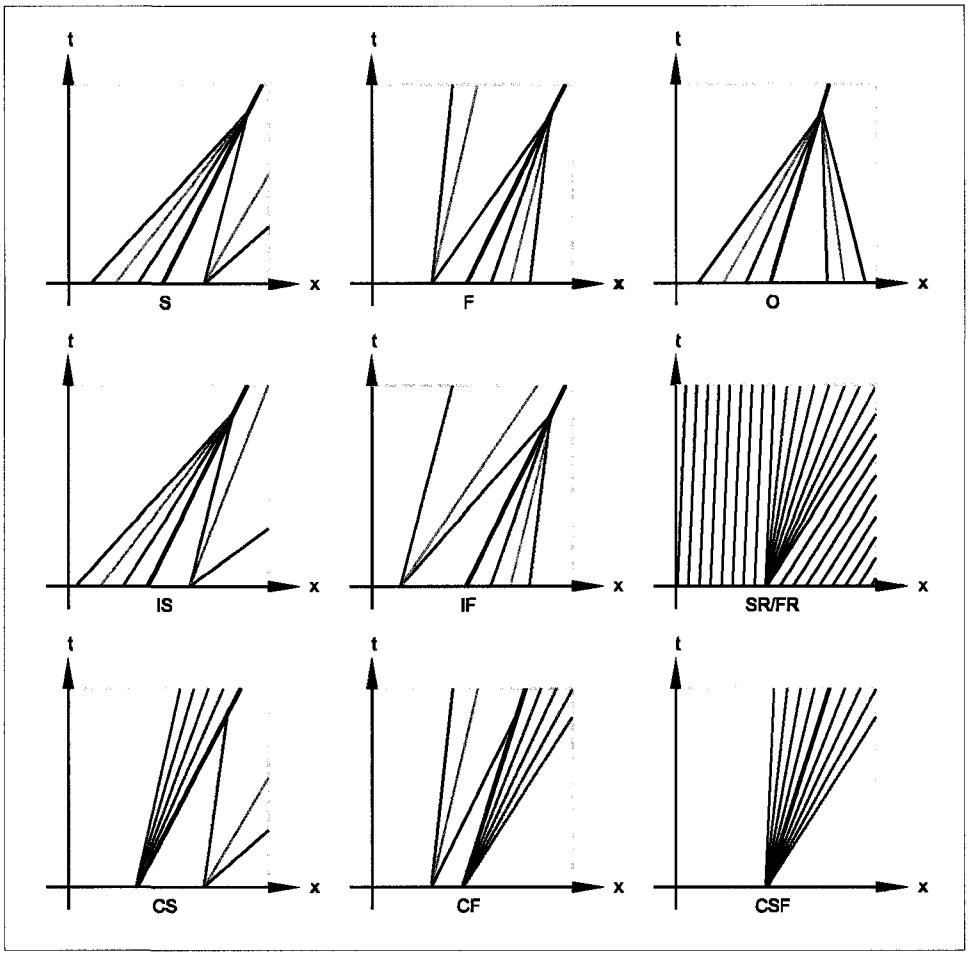


Figure 1-5: Viscous Shocks and rarefactions classified by characteristics (Myong and Roe, 1997b). Thin lines are characteristics, red coresponds to slow magnetosonic, green to Alfven, and blue to fast magnetosonic. Thin gray lines represent rarefaction fans. Thick dark line represent the shock.

direction(relative to the shock interface) while there is no jump in the normal component of the magnetic flux density. Thus in both cases the tangential magnetic field retains its direction (Jeffrey, 1966, p. 214). In the third column of the first row we have the situation identifying overcompressive shocks where all three of the magnetoacoustic characteristics converge into the shock. Transverse shocks(also known as rotational discontinuities), characterized by jumps in the transverse components of the magnetic flux density and velocity with pressure and density constant crossing the shock, are of this type (Myong and Roe, 1997a). The strength of the magnetic field is unchanged across a transverse shock, while the transverse field is rotated on crossing plane of the shock (Jeffrey, 1966, p. 216). For exam-

ple in Figure 5-1 the plot of B_y vs. B_z shows the rotational discontinuities clearly. On the second row the first two cases shown correspond to the Intermediate slow and fast shocks. These are characterized by the convergence of the Alfven characteristic and respectively one of the slow or fast magnetosonic characteristics. Unlike the fast and slow Lax type shocks, the sign of the transverse magnetic field may change across the intermediate fast and slow shock (Myong and Roe, 1997b). All three columns of the third row corespond to compound waves. For all compound waves the shock speed, **s**, is the same as one of the characteristic speeds on one side of the discontinuity. The compound shocks can form where at a point on a Hugoniot curve, the Hugoniot curve is tangent to an integral curve and the shock speed is equal to the characteristic speed associated with the integral curve (Myong and Roe, 1997a). More will be said in regard to compound waves after the introduction of the rarefaction fan in Section 1.12. Finally, all shocks are compressive and pressure increases from downstream to upstream across the shock. Additionally the jump in density across any given shock is limited to

$$\rho^L < \rho^R \frac{\gamma + 1}{\gamma - 1} \tag{1.75}$$

where L and R identify up and down stream states (Jeffrey, 1966) p 208.

In summary one can say that given the complex eigenstructure of the ideal MHD system the decisions one can make about shock admissibility are restricted by the limited amount of physics contained therein. By use the Lax entropy condition, or even the evolutionary admissibility condition, in order to decide which shocks are physically admissible, one discards some of the physically valid cases. By using the viscous admissibility condition one makes a more informed decision as to which shocks are physically admissible since the viscous admissibility condition has been derived in a context where more physics have been accounted for in the MHD equations. With that said there are known cases where MHD Riemann problems do not have a unique solution under the viscosity admissibility condition, and so the question of shock admissibility has not yet been conclusively settled (Torrilhon, 2003).

1.12 Rarefaction Fans

Generally speaking two arbitrary adjacent states cannot always be connected by a shock, since shocks are always compressive. Connecting two states across a discontinuity at times requires a decrease in pressure. Smooth transitions from high, up stream, to low, down stream, pressure are called rarefactions. The solutions of the Riemann problem all have the property that they are constant along a ray $x = \lambda t$. By making the substitution $\mathbf{u} = \mathbf{w}(\xi = x/t)$ it can be shown that given some known state $\mathbf{u}^L$, the states $\mathbf{u}^R$ that can be paired with it to form a solution satisfying the second law of thermodynamics where pressure decreases from downstream to upstream, all lie on a single monotonically increasing integral curve of the eigenvector field parametrized by λ_i (LeVeque, 1992) p 83. Thus integral curves are to rarefactions as Hugoniot curves are to shocks. For strictly hyperbolic systems there are p integral curves through each point in phase space, one corresponding to each eigenvector of the system. A property of integral curves is that they are everywhere tangent to their respective eigenvectors. Given some known state $\mathbf{u}^L$, finding $\mathbf{u}^R$ across the rarefaction is accomplished by solving a differential equation of the form (LeVeque, 1992) p 84,

$$\mathbf{w}'(\xi) = \frac{\mathbf{r}_i(\mathbf{w}(\xi))}{\nabla\lambda_i(\mathbf{w}(\xi)) \cdot \mathbf{r}_i(\mathbf{w}(\xi))}, \qquad \xi_0 \leq \xi \leq \xi_1 \tag{1.76}$$

for each characteristic field. Here the $\mathbf{r}_i$ are the right eigenvectors. For genuinely nonlinear systems $\lambda_i(\mathbf{u})$ always varies monotonically on the integral curves of $\mathbf{r}_i(\mathbf{u})$ by definition. For systems which are not genuinely nonlinear states $\mathbf{u}^R$ that can be connected by a rarefaction to $\mathbf{u}^L$ all lie between $\mathbf{u}^L$ and the first maxima of the integral curve parametrized with respect to λ_i (LeVeque, 1992, p. 84).

As mentioned above compound shocks form when at a critical point on a Hugoniot curve the Hugoniot curve is tangent to an integral curve, and the shock speed is equal to the characteristic speed associated with the integral curve. It is because these shocks are coupled to a rarefaction they are called compound. There are three cases which can occur. These cases are depicted on the third row in Figure 1-5. First, the slow compound wave consisting of a rarefaction wave followed by a slow shock. Second, the fast compound wave

consisting of a fast shock followed by a fast rarefaction wave. Finally, the fast slow compound wave where at a point a Hugoniot curve is tangent to both slow and fast rarefaction, resulting in a slow rarefaction followed by a jump to a fast rarefaction. In all of these cases the shock speed is given by the characteristics adjacent to the shock (Myong and Roe, 1997a).

1.13 Constructing Solutions to the Riemann Problem

In general one can construct a solution the Riemann problem for two initial states $\mathbf{u}^L$ and $\mathbf{u}^R$ by connecting series of intermediate states through shocks, contact discontinuities and rarefactions. Such a solution is a specific union of admissible Hugoniot curves, contacts and locally monotonically increasing integral curves. Riemann problems are possibly the best validation for numerical code since they can be devised such that their solutions contain a rich structure, encompassing many key physical principles of the ideal MHD system. Additionally, the discontinuous features characteristic of solutions to Riemann problems often lead to the break down of numerical finite difference methods. One would like to be sure that if shocks are expected in some scientific modeling application that the numerics employed will not break down in that parameter space. Most importantly, we have the capability to find an exact solution to the Riemann problem. This is a critical point in MHD code development which many over look. For example in many publications on MHD numerics a comparison of the numerical solutions of MHD Riemann problems with exact solutions are not shown, instead an external paper with published numerical solutions is cited and it is claimed that the numerical results presented in a series of compact figures are in good qualitative agreement. This is a highly deficient practice! The problem is that solution features such as shock front location and jump magnitudes are very difficult to precisely determine by eye. Making matters worse, the potential presence of subtle software errors can lead to results that look good but are just slightly off. A number of published exact solutions which can be used for code validation are available for instance, (Torrilhon, 2002), (Torrilhon, 2003), (Wu, 1995), and, (Kim et al., 1999).

CHAPTER 2

NUMERICAL METHODS FOR IDEAL MHD

2.1 Chapter Abstract

In this chapter a review of shock capturing numerical methods for nonlinear hyperbolic systems is presented. Numerical conservation is emphasized for its importance where shocks and discontinuities are concerned. A discussion of nonlinear numerical stability is presented along with my analysis of the numerical sources of instability inherent in high-order numerical methods and a physical interpretation of unconditional stability of a class of lower order numerical methods. I will present the MHD Constrained Transport method which I have used in my code to ensure that the divergence free constraint is numerically satisfied. I will then present my analysis of the numerical properties of a specific implementation of the Constrained Transport method from which I derive an upper bound on the order of the growth rate of the divergence of the magnetic flux density. I will discuss the coupling of the constrained transport grids to the semi-conservative discretization and briefly discuss how the semi-conservative formulation can be used to alleviate the numerical issue of maintaining pressure positivity in accordance with the laws of thermodynamics. The discussion in this chapter lays the foundation for my development of a new Constrained Transport method for the block structured adaptive mesh methods which are presented in chapter. 3.

2.2 Finite Difference Methods for Conservation Laws

In order to compute a numerical solution to systems of the form (1.14) we will sample the fields of $\mathbf{u}$ given at time, t_n, at discrete spatial locations distributed over a box shaped domain, and discretize both the temporal and spatial partial derivatives in (1.14). The so-

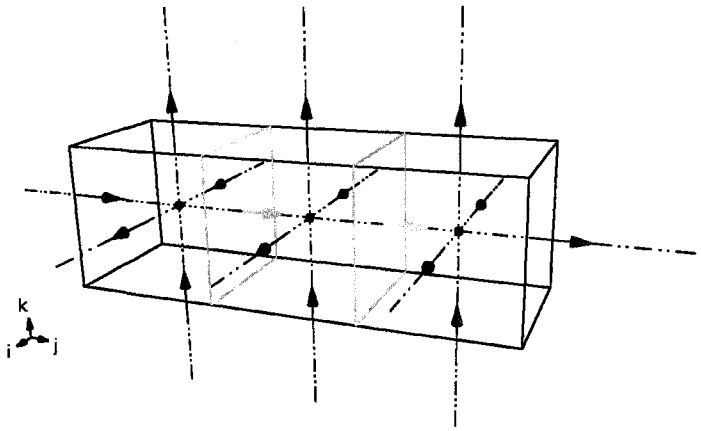


Figure 2-1: Three Finite volume computational cells. Colored dots indicate cell center, colored arrows indicate face normal fluxes. Shared faces and fluxes are colored teal.

lution of the system of partial differential equations will be evolved in time by the repetition of many small, discrete, incremental steps until the solution is obtained at the desired end time. It is unlikely (for reasons to be introduced below) that the time increment of these steps will be uniform. Therefor the time after a single intermediate incremental step will be given by:

$$t_{n+1} = t_n + \Delta t_n \tag{2.1}$$

where:

$$t_n = t_0 + \sum_{i=1}^{n-1} \Delta t_i. \tag{2.2}$$

We choose a Cartesian coordinate system and define a cubical finite volume(or computational cell), V, on which we will apply equation (1.13). Each finite volume is uniquely identified by a three integers, i, j, k, therefore we will introduce the notation $V_{i,j,k}$. Any computational cell so identified will be spatially located such that its center is coincident with the point:

$$\mathbf{x}(i,j,k) = \Delta\mathbf{x} \cdot (i,j,k) + \mathbf{x}_0, \tag{2.3}$$

where $\mathbf{x}_0$ is the cell center of the cell in the lower left corner of the computational domain. At time any discrete time, t_n, the discrete state variables,

$$\mathbf{u}^n_{ijk} \equiv (u^n_{1,ijk}, u^n_{2,ijk}, ..., u^n_{p,ijk})^T, \tag{2.4}$$

are constant on a given finite-volume, V_{ijk}. The surface S bounding the finite-volume is composed of the union of the 6 faces: $S_{i\pm\frac{1}{2},jk}$,$S_{i,j\pm\frac{1}{2},k}$ and $S_{ij,k\pm\frac{1}{2}}$. According to (1.13) we need the fluxes, evaluated on computational cell faces. We introduce the following equivalent notations for the physical fluxes,

$$\mathbf{f}\left(\mathbf{u}_{ijk}^{n}\right) \equiv \mathbf{F}(\mathbf{u}) \equiv (\mathbf{f}(\mathbf{u}), \mathbf{g}(\mathbf{u}), \mathbf{h}(\mathbf{u})), \tag{2.5}$$

that will be used interchangeably depending on the context. Due to the above restriction that the solution take a constant value on each computational cell, with the exception of the outer boundary of the problem domain, the physical fluxes, $\mathbf{f}\left(\mathbf{u}_{i,j,k}^{n}\right)$, are double valued at the computational cell faces. Rather than use them directly we will compute a corresponding numerical flux on one of the faces of S, $S_{i\pm\frac{1}{2},jk}$, $S_{i,j\pm\frac{1}{2},k}$ and $S_{ij,k\pm\frac{1}{2}}$, by the numerical flux functions:

$$\begin{pmatrix} \hat{\mathbf{f}}_{i+\frac{1}{2},jk}^{n}(..., (\mathbf{f}(\mathbf{u}_{i-1,j,k}^{n}), \mathbf{f}(\mathbf{u}_{ijk}^{n}), \mathbf{f}(\mathbf{u}_{i+1,j,k}^{n})...), \\ \hat{\mathbf{g}}_{i+\frac{1}{2},jk}^{n}(..., (\mathbf{g}(\mathbf{u}_{i,j-1,k}^{n}), \mathbf{g}(\mathbf{u}_{ijk}^{n}), \mathbf{g}(\mathbf{u}_{i,j+1,k}^{n})...), \\ \hat{\mathbf{h}}_{i+\frac{1}{2},jk}^{n}(..., (\mathbf{h}(\mathbf{u}_{i,j,k-1}^{n}), \mathbf{h}(\mathbf{u}_{ijk}^{n}), \mathbf{h}(\mathbf{u}_{i,j,k+1}^{n})...), \end{pmatrix} \tag{2.6}$$

which amounts to some combination of the nearby discrete fluxes. Using this notation we can apply the fundamental theorem of calculus and partially discretize the integral form of the conservation law:

$$\iiint_V \mathbf{u}(\mathbf{x}, t_0 + dt) - \mathbf{u}(\mathbf{x}, t_0)\, dV + \int_{t_0}^{t_0+dt} \iint_S \mathbf{f}(\mathbf{u}) \cdot dS\, dt = 0, \tag{2.7}$$

to get:

$$\left(\frac{\partial}{\partial t}\mathbf{u}_{ijk}\right)\Delta V_{ijk} + \left(\hat{\mathbf{f}}_{i+\frac{1}{2},jk}^{n} - \hat{\mathbf{f}}_{i-\frac{1}{2},jk}^{n}\right)\Delta S_{i+\frac{1}{2},jk} + \left(\hat{\mathbf{g}}_{i,j+\frac{1}{2},k}^{n} - \hat{\mathbf{g}}_{i,j-\frac{1}{2}.k}^{n}\right)\Delta S_{i,j+\frac{1}{2},k} + \left(\hat{\mathbf{h}}_{ij,k+\frac{1}{2}}^{n} - \hat{\mathbf{h}}_{ij,k-\frac{1}{2}}^{n}\right)\Delta S_{ij,k+\frac{1}{2}} = 0$$

where ΔV_{ijk} is the computational cell volume, $\Delta S_{i\pm\frac{1}{2},jk}$ are the surface areas of the faces $S_{i\pm\frac{1}{2},jk}$, and $\hat{\mathbf{f}}$, $\hat{\mathbf{g}}$, and $\hat{\mathbf{h}}$ are the components of the numerical flux. We have not discretized the time derivative to highlight the fact that there is a good deal of flexibility in this regard. Multiplying by $(\Delta V_{ijk})^{-1}$ we obtain the semi-discrete conservative differential form:

$$\frac{\partial}{\partial t}\mathbf{u}_{ijk} +$$

$$\frac{\hat{\mathbf{f}}^n_{i+\frac{1}{2},jk} - \hat{\mathbf{f}}^n_{i-\frac{1}{2},jk}}{\Delta x} + \frac{\hat{\mathbf{g}}^n_{i,j+\frac{1}{2},k} - \hat{\mathbf{g}}^n_{i,j-\frac{1}{2},k}}{\Delta y} + \frac{\hat{\mathbf{h}}^n_{ij,k+\frac{1}{2}} - \hat{\mathbf{h}}^n_{ij,k-\frac{1}{2}}}{\Delta z} = 0 \tag{2.8}$$

This expresses, for a single computational cell, that the rate of change of $\mathbf{u}_{ijk}$ is a function the flux through its surface. Thus, on a single cell the conservation principle is satisfied. However, in terms of numerical finite difference methods one needs to not only satisfy conservation principle on a single cell but also on any aggregate collection of adjacent cells when the calculations are made cell by cell. The situation is depicted in Figure 2-1, where an aggregate of three cells is shown. An expression equating the time rate of change of the solution on an arbitrary aggregate collection of adjacent cells, cell by cell is given by:

$$\begin{aligned}\frac{\partial}{\partial t}\sum_{i=a}^{p}\sum_{j=b}^{q}\sum_{k=c}^{r}\mathbf{u}_{ijk} = \sum_{i=a}^{p}\sum_{j=b}^{q}\sum_{k=c}^{r}\Bigg[&\frac{\Delta t}{\Delta x}\left(\hat{\mathbf{f}}^n_{i+\frac{1}{2},jk} - \hat{\mathbf{f}}^n_{i-\frac{1}{2},jk}\right)\\ &+\frac{\Delta t}{\Delta y}\left(\hat{\mathbf{g}}^n_{i,j+\frac{1}{2},k} - \hat{\mathbf{g}}^n_{i,j-\frac{1}{2},k}\right) + \frac{\Delta t}{\Delta z}\left(\hat{\mathbf{h}}^n_{ij,k+\frac{1}{2}} - \hat{\mathbf{h}}^n_{ij,k-\frac{1}{2}}\right)\Bigg],\end{aligned} \tag{2.9}$$

while application of the conservation principle to the volume enclosing the same collection of cells as a whole is given by:

$$\begin{aligned}\frac{\partial}{\partial t}\sum_{i=a}^{p}\sum_{j=b}^{q}\sum_{k=c}^{r}\mathbf{u}_{ijk} = &\frac{\Delta t}{\Delta x}\sum_{j=b}^{q}\sum_{k=c}^{r}\left(\hat{\mathbf{f}}^n_{p+\frac{1}{2},jk} - \hat{\mathbf{f}}^n_{a-\frac{1}{2},jk}\right)\\ &+\frac{\Delta t}{\Delta y}\sum_{i=a}^{p}\sum_{k=c}^{r}\left(\hat{\mathbf{g}}^n_{i,q+\frac{1}{2},k} - \hat{\mathbf{g}}^n_{i,b-\frac{1}{2},k}\right) + \frac{\Delta t}{\Delta z}\sum_{i=a}^{p}\sum_{j=b}^{q}\left(\hat{\mathbf{h}}^n_{ij,r+\frac{1}{2}} - \hat{\mathbf{h}}^n_{ij,c-\frac{1}{2}}\right).\end{aligned} \tag{2.10}$$

For the conservation principle to be numerically satisfied, the right hand sides of (2.9) and (2.10) must be equivalent. This occurs when:

$$\begin{aligned}\hat{\mathbf{f}}^n_{i+\frac{1}{2},jk} &= \hat{\mathbf{f}}^n_{(i+1)-\frac{1}{2},jk}\\ \hat{\mathbf{g}}^n_{i,j+\frac{1}{2},k} &= \hat{\mathbf{g}}^n_{i,(j+1)-\frac{1}{2},k}\\ \hat{\mathbf{h}}^n_{ij,k+\frac{1}{2}} &= \hat{\mathbf{h}}^n_{ij,(k+1)-\frac{1}{2}}\end{aligned} \tag{2.11}$$

Therefore, a numerical method is said to be conservative only if the divergence of its fluxes can be written in the form:

$$\nabla\cdot\mathbf{f}(\mathbf{u}) \approx \frac{\hat{\mathbf{f}}^n_{i+\frac{1}{2},jk} - \hat{\mathbf{f}}^n_{i-\frac{1}{2},jk}}{\Delta x} + \frac{\hat{\mathbf{g}}^n_{i,j+\frac{1}{2},k} - \hat{\mathbf{g}}^n_{i,j-\frac{1}{2},k}}{\Delta y} + \frac{\hat{\mathbf{h}}^n_{ij,k+\frac{1}{2}} - \hat{\mathbf{h}}^n_{ij,k-\frac{1}{2}}}{\Delta z} \tag{2.12}$$

such that (2.11) holds.

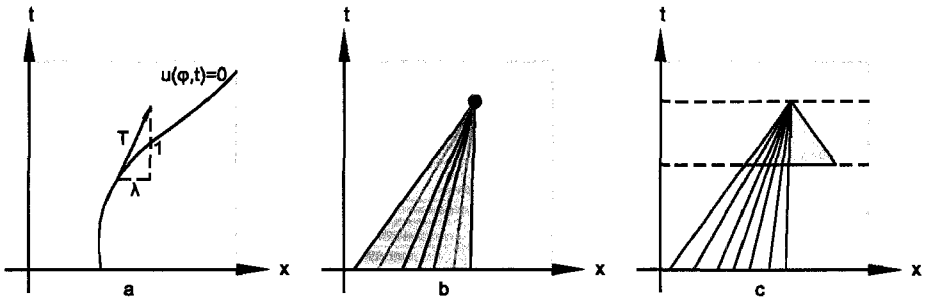


Figure 2-2: Numerical and physical domain of dependence. a: Tangent to a 1 dimensional characteristic surface is shown. The tangnet vector is decomposed into its componets λ and 1. b: The physical domain of dependence(light blue region) of a given point in x, t space is found by tracing backward along characteristic surfaces. Blue coresponds to the fast characteristic, green to the Alfven, red to the slow, black to flow speed. c: The numerical domain of dependence(light pink region) must contain the physical domain of dependence.

As discussed above, the conservation form is key when shocks are expected to be a part of the solution because it has been shown that non-conservative schemes can converge to the wrong solution if the solution contains a shock (Hou, 1994), and additionally the Rankine-Hugoniot conditions would not necessarily be satisfied across shocks (Brackbill and Barnes, 1989). Deviations from the conservation form should be justified as inevitably the result is that finite difference schemes can mislocate shock fronts and miscalculate jump magnitudes across them. However, the situation must be examined on a case by case basis, as a non conservative form may perform well enough for a given application.

2.3 The CFL Condition

A conservative finite difference method is called central if the individual components of the numerical flux functions have the form:

$$\begin{pmatrix} \hat{\mathbf{f}}^n_{i+\frac{1}{2},jk}((\mathbf{f}(\mathbf{u}^n_{i-\kappa,j,k}), ..., \mathbf{f}(\mathbf{u}^n_{i+\kappa,j,k})), \\ \hat{\mathbf{g}}^n_{i+\frac{1}{2},jk}((\mathbf{g}(\mathbf{u}^n_{i,j-\kappa,k}), ..., \mathbf{g}(\mathbf{u}^n_{i,j+\kappa,k})), \\ \hat{\mathbf{h}}^n_{i+\frac{1}{2},jk}((\mathbf{f}(\mathbf{u}^n_{i,j,k-\kappa}), ..., \mathbf{h}(\mathbf{u}^n_{i,j,k+\kappa})), \end{pmatrix} \tag{2.13}$$

Where the set of cells,

$$\{V_{i-\kappa,j,k}, ..., V_{i+\kappa,j,k}, V_{i,j-\kappa,k}, ..., V_{i,j+\kappa,k}, V_{i,j,k-\kappa}, ..., V_{i,j,k+\kappa}\}, \tag{2.14}$$

from which the numerical flux is computed from is called the numerical domain of dependence and κ is called the stencil width. Although schemes where the stencil is given an upwind bias are quite popular, I will only consider central schemes here.

If we fix a point and look backward in time then as previously mentioned its current state can only have been influenced by points in its physical domain of dependence(see Figure 2-2.b). Therefor, when advancing a finite difference equation in time one must ensure that the numerical domain of dependence contains the physical domain of dependence if we expect the result to approximate the true solution. This is known as the CFL condition. If the CFL condition is not satisfied the numerical approximation has not taken into account all of the characteristic wavefronts who's interference at a given point in space and time constitute the solution. Because the numerical domain of dependence for a given finite difference formula is fixed by its stencil, one must make an adjustment to the time step to satisfy the CFL condition. The length of the base of the physical domain of dependence of the evolved point u_i^{n+1} is given by: $D_p = 2|\lambda|\Delta t$, where λ is the fastest characteristic speed. This must be less than or equal to the legnth of numerical domain of dependence, which for a central scheme is given by: $D_n = 2\kappa\Delta x$. Combining these into an expression for Δt

$$\Delta t = \frac{\alpha\kappa\Delta x}{|\lambda_{max}|} \tag{2.15}$$

Here λ_{max} is the fastest characteristic speed. In the case of ideal MHD this is given by (1.55). For simplicity note that (1.55) is maximized perpendicular to the magnetic field so that an upper bound is given by:

$$\lambda_{max} = \max\left(\mathbf{v} \pm \sqrt{\frac{\mathbf{B}\cdot\mathbf{B}}{\rho} + a^2}\right) \geq \lambda_{7,8}. \tag{2.16}$$

Prior to each finite difference timestep we will employ (2.15) to calculate the largest Δt that will ensure that the CFL condition is satisfied. A safety factor, known as the CFL number, given by α in (2.15) where $0 \leq \alpha \leq 1$, is often included in the time step size calculation. An important point to consider when implementing numerical methods is that the error in the solution for a single finite difference step, the local truncation error(LTE), is on the order

of:

$$\xi \equiv O\left(\Delta x^n, \Delta t^q\right) \tag{2.17}$$

where n is the order of the spatial finite difference method, and q the order of the temporal finite difference method. However these errors tend to accumulate and the global error grows as a function of the number of finite difference steps taken. A crude estimate of the overall error is: $m\xi$, where m is the number of finite difference steps taken. Given that global errors grow as a function of the number of finite difference steps taken, and that growth rate to a crude approximation is proportional to the truncation error, we would like to compute the solution minimizing both number of steps and the truncation error. For these reasons numerical methods of order 2 and higher are typically used. We should be conscious that decreasing the grid spacing too much can also have a negative effect. At some point the order of the local truncation error will be comparable to the rounding error and then rounding errors will dominate. See (Burden and Faires, 2001, p. 173) for some examples.

2.4 Shock Capturing Finite Difference Methods and Numerical Stability

Numerical finite difference methods which work in the presence of shocks are known as shock capturing methods. Shock capturing methods must be conservative, and numerically stable in the presence of shocks and contacts where the derivative and consequently the finite difference formula itself is undefined. The term numerical instability is used to characterize the unbounded (or unacceptable) growth of small disturbances which a given finite-difference method introduces in the approximate solution and amplifies over time. A discretization or numerical method is unstable if it produces large errors that increase with time, which are significantly larger than the discretization error. Otherwise the method is stable (Laney, 1998, p. 272). Generally speaking nonlinear stability analysis of multi-dimensional systems proves difficult. However, many useful results can be derived in a one

dimensional setting. The bulk of nonlinear stability theory has been developed specifically with the one dimensional equations of gasdynamics in mind. These results are based on the range diminishing property, or one of its relatives, such as the total variation diminishing (TVD) property, or the positivity property. The range diminishing property is a physical property of the solutions to the one dimensional gasdynamic equations, namely that: existing minima either stay the same or increase while existing maxima either stay the same or decrease and over time no new maximum or minimum are introduced, except during the break up of unstable discontinuities present in the initial conditions or as a result of shock to shock interaction (Laney, 1998, p. 274). The MHD equations are quite a bit different than the gasdynamic equations, however the physical characteristics of their solutions are quite similar. For example the exact solution to the MHD Riemann problem shown in Figure 1-4 illustrates the range diminishing property quite well, note the absence of extrema. For both systems, the presence of shocks in the solution are the primary concern where nonlinear stability of numerical methods is concerned. If a numerical scheme can be shown to produce approximate solutions having the range-diminishing property then the scheme is said to be nonlinearly stable in the presence of discontinuities.

Often to simplify things a numerical method is analyzed using the one dimensional scalar conservation equation,

$$\frac{\partial u_i}{\partial t} + \frac{\partial f(u_i)}{\partial x} = 0. \tag{2.18}$$

who's solution, like the one dimensional gasdynamic equations, has the range diminishing property. If we can show that a given finite difference method has the range diminishing property when applied to (2.18) then it also has the range diminishing property when applied to one dimensional gasdynamic system of equations (Laney, 1998, p. 274). Thus, much of the nonlinear stability theory stems from the study of scalar conservation equation rather than the equations of gasdynamics or ideal MHD themselves. The range-diminishing property is only true in the one dimensional case. However, multidimensional numerical methods which satisfy the range diminishing property in one dimension are nonlinearly stable in multiple dimensions. The primary concern with this extension from a single to multiple dimensions

is that the one-dimensional conditions may be too strict in the multidimensional case since the true multi-dimensional solutions do not actually have the range diminishing property.

A related nonlinear stability condition, implied by the range diminishing property, is the TVD condition. A numerical method is said to be TVD in a single dimension if:

$$TV(u^n) \leq TV(u^{n+1}) \tag{2.19}$$

for all, n, where

$$TV(u^n) = \sum_{i=-\infty}^{\infty} |u_{i+1}^n - u_i^n|. \tag{2.20}$$

The total variation is rough estimate of the amount of oscillation in a solution. The TVD condition says that total variation must either stay the same or decrease with each finite-difference step.

The finite difference methods that have been implemented in the code shown here in their conservation form are, Rusanov's scheme (Rusanov, 1962), (Sod, 1978),

$$\hat{f}_{i+\frac{1}{2},jk}^n = \frac{1}{2}(f_{ijk}^n + f_{i+1,jk}^n) - \frac{1}{4}(\alpha_{ijk} + \alpha_{i+1,jk})(u_{i+1,jk}^n - u_{ijk}^n), \tag{2.21}$$

where, $\alpha_i = (|v_i| + a_i)$, is an estimate of the fastest hydrodynamic characteristic speed, and, Zalesak's forth order central scheme (Zalesak, 1981), (Zalesak, 1984),

$$\hat{f}_{i+\frac{1}{2},jk}^n = \frac{7}{12}(f_{ijk}^n + f_{i+1,jk}^n) - \frac{1}{12}(f_{i-1,jk}^n + f_{i+2,jk}^n). \tag{2.22}$$

In both formulas the numerical fluxes are differentiated from the physical fluxes by a hat.

It is common practice to present a finite-difference method in its conservation form, however this tends to obfuscate some of its properties. An algebraic manipulation placing the scheme in a more familiar form will reveal its origin, and give insight into its properties. First consider the Zalesak scheme. We will consider a single component, namely $\hat{\mathbf{f}}$, as the analysis for other two is identical. Computing its contribution to the divergence we find that,

$$\frac{\partial f(u_i)}{\partial x} \approx \frac{f_{i-2} - 8f_{i-1} - 8f_{i+1} - f_{i+2}}{12\Delta x}, \tag{2.23}$$

where the subscripts j and k have been dropped from convenience because they do not vary here. This is the centered 5 point formula which is derived by taking the derivative of a

quartic Lagrange interpolant (Burden and Faires, 2001, p. 171). Lagrange interpolants form the basis of many popular finite difference discretizations. For Lagrange based methods the local truncation error, is given by,

$$f(x) = P(x) + \frac{f^{(n+1)}(\xi)}{(n+1)!}\prod_{i=0}^{n}(x - x_i). \tag{2.24}$$

However, the specific use of this formula is restricted to those functions whose derivatives have known bounds (Burden and Faires, 2001). In the neighborhood of a strong discontinuity the bound on the error of Lagrange based finite difference formulas is undefined. Near the discontinuity the interpolant from which it is constructed tends to oscillate wildly leading to the introduction of spurious oscillations in the solution and inherent numerical instability near the discontinuity.

A plot of a solution containing a discontinuity is shown in Figure 2-3 along with the derivative approximated by the Zalesak scheme evaluated at four consecutive points. Circle markers indicate cell centers and each cell is given an identifying color. The gray lines are plots of the quartic Lagrange interpolants that are being used in each of the differentiations. These have been colored and thickened near the point which they are associated with. Squares indicate the value of the computed value of the derivative at the given point. From this plot we can see there are large errors appearing in the derivative forming as soon as one of the interpolating points crosses the discontinuity. If we mentally advance the solution using the scalar conservation equation we can see that three extrema are introduced as a result of the oscillatory errors in the interpolants that were used to compute the derivative. Unfortunately the spurious oscillations thus introduced are known to grow in time and destroy the solution eventually.

Now consider the Rusanov scheme. Again we will consider the single component, $\hat{\mathbf{f}}$. Let $\alpha_{i+\frac{1}{2}} = \alpha_{i+1} + \alpha_i$ making the substitution we have:

$$\frac{\partial \mathbf{f}(u_i)}{\partial x} \approx \frac{\mathbf{f}_{i+1} - \mathbf{f}_{i-1}}{2\Delta x} - \frac{\alpha_{i+\frac{1}{2}}(u_{i+1} - u_i) - \alpha_{i-\frac{1}{2}}(u_i - u_{i-1})}{4\Delta x} \tag{2.25}$$

Further, for the sake of a simple demonstration, let $\alpha_{i+1/2} = \alpha_{i-1/2} = 4\epsilon/\Delta x$ then we have,

$$\frac{\partial f(u_i)}{\partial x} \approx \frac{f_{i+1} - f_{i-1}}{2\Delta x} - \epsilon\left(\frac{u_{i+1} - 2u_i + u_{i-1}}{(\Delta x)^2}\right). \tag{2.26}$$

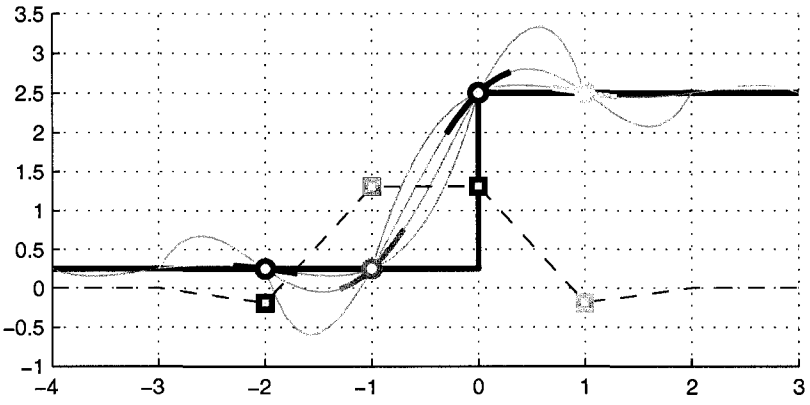


Figure 2-3: Why finite difference methods break down near a discontinuity. Initial condition u_i(thick black line) containing a discontinuity at $i = 0$ shown with Zalesak's finite difference approximation of the derivative(dashed thin line)

The first term here is a centered 3 point formula which is derived by taking the derivative of a quadratic Lagrange interpolant (Burden and Faires, 2001, p. 171). This term gives a second order approximation. However if used alone would experience the issues similar to those illustrated above for the Zalesak scheme. Note that the second term in (2.26) has the form of a three point Lagrange interpolant based formula approximating the second derivative (Burden and Faires, 2001, p. 173). If we substitute this, and similar expressions for for derivatives in the other coordinate directions, into our conservation equation we will obtain an expression of the form:

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \mathbf{f}(\mathbf{u}) = \epsilon \nabla^2 \mathbf{u}. \tag{2.27}$$

This is of the viscous/diffusive form discussed in previously in the section regarding shocks. Viscous terms are typically small where the solution is smooth only becoming significant near discontinuities tending to give the discontinuity a smooth, so called, viscous profile. We are not solving the diffusive form of the MHD equations, but our finite difference scheme is formulated as if we were. The term $\epsilon \nabla^2 \mathbf{u}_{ijk}$ is thus called an artificial viscosity term (Laney, 1998, p. 251). A stabilizing effect is achieved by such methods, allowing their use near discontinuities in the solution. However, the magnitude of the coefficient of artificial viscosity is most likely much larger than that of a real gas the scales where the viscous

physics operate are orders of magnitude smaller than we will typically be willing to resolve. All this has the effect that the viscous profile introduced at a discontinuity due to artificial viscosity is much thicker than that which would occur in a real gas. Unless we highly resolve the shock and use the appropriate coefficient of viscosity the accuracy of the solution is reduced. Thus, the stability attained with artificial viscosity methods comes at a price, namely a reduction of order locally in the regions of a discontinuity. I have made a number of gross over simplifications in the preceding manipulations to show that Rusanov's scheme is an artificial viscosity type scheme. Specifically I made use of a somewhat arbitrary value for $\alpha_{i\pm1/2}$'s. However, it turns out that the nonlinear stability conditions known as positivity can be directly applied to Rusanov's scheme as written in equation (2.25) (Laney, 1998, p. 288), (Tadmor, 1984). It should be noted that the positivity conditions restrict κ in the equation (2.15) to 1 regardless of the stencil width of the scheme employed, and additionally in three dimensions the CLF number α is restricted to less than or equal to 1/3 (Toro, 1997) p 526. These conditions can be used to show that the Rusanov scheme satisfies the TVD property, thus the Rusanov scheme is non-linearly stable. Information regarding non-linear numerical stability and its extension to multi-dimensions can be found in, (Tadmor, 1984), (Lax, 1997), (Harten, 1983), (Liu and Lax, 1996), (Goodman and LeVeque, 1985).

One would like to have the higher order accuracy of the high order Lagrange based finite difference formula, while attaining the stability of the artificial viscosity formula. The desired affect can be achieved by applying the high order scheme where the solution is smooth and the low order scheme near discontinuities in the solution. In a conservative formulation this is achieved by:

$$\hat{\mathbf{f}}^n_{i+\frac{1}{2}} = \theta_{i+\frac{1}{2}}\hat{\mathbf{f}}^n_{i+\frac{1}{2},TVD} + (1-\theta_{i+\frac{1}{2}})\hat{\mathbf{f}}^n_{i+\frac{1}{2},H} \tag{2.28}$$

where $0 \leq \theta \leq 1$. In the first term the subscript TVD indicates that a non-linearly stable scheme is used, and the subscript H indicates that a high order Lagrange based scheme is used. This is called a hybrid flux. Near discontinuities and extrema $\theta = 1$ while in regions where the solution is smooth $\theta = 0$. Thus θ is called a shock switch or flux limiter. Any expression that achieves the desired result is acceptable and the construction of such

formula is as much an art as a science. We will define $\theta_{i+12} = \max(R_{i-1}, R_i, R_{i+1}, R_{i+2})$ where R is given by Raeder's limiter:

$$R_i = \begin{cases} \left[\dfrac{||u_i - u_{i-1}| - |u_{i+1} - u_i||}{r\left(|u_{i+1}| + |u_i| + |u_{i-1}|\right) + |u_{i+1} - u_i| + |u_i - u_{i-1}| + \epsilon}\right]^4, & \delta_f \delta_b \geq 0 \\ 1, \quad \delta_f \delta_b < 0 \end{cases} \tag{2.29}$$

where $\delta_f = u_{i+1} - u_i$, $\delta_b = u_i - u_{i-1}$. ϵ, a small number on the order of 10^{-10}, is present to ensure a non-zero denominator and r is typically a small number on the order of 10^-3.

2.5 The Constrained Transport Method

As we have seen in earlier sections the constraint equation isn't precisely satisfied in numerical methods such that the divergence grows as a function of the number of finite difference steps taken. We must take precautions so that this growth does not destroy the solution over the time period of interest. To that end we will take a close look at the situation on the finite volume enclosed by a computational cell. Specifically, the magnetic flux through the surface of one computational cell is given by

$$\Phi \equiv \iint \mathbf{B} \cdot d\vec{\mathbf{S}}. \tag{2.30}$$

By virtue of Stokes's Theorem we can write the time change of Φ for each cell face as:

$$\frac{\partial \Phi}{\partial t} = \iint \frac{\partial \mathbf{B}}{\partial t} \cdot d\vec{S} = \iint (-\nabla \times \mathbf{E}) \cdot d\vec{S} = -\oint \mathbf{E} \cdot d\vec{\ell}. \tag{2.31}$$

We discretize these integrals by placing components of the magnetic flux density on cell faces, and components of $\mathbf{E}$ on the face edges, as shown in Figure 2-4. Here, and in the figures to follow, we introduce a general Cartesian coordinate system in the orthogonal basis vectors $\hat{\Theta}$,$\hat{\Psi}$,$\hat{\Omega}$. Discretized quantities are identified by the subscripts ijk. These subscripts when used with $\mathbf{B}$ and $\mathbf{E}$, identify the base face or base edge location respectively. Integer subscripts give the relative location with respect to the base edge or face; for example, $010 \equiv i, j+1, k$. Cell edge lengths are given by $\Delta\Theta$, $\Delta\Psi$, $\Delta\Omega$. Discretizing (2.31), we have

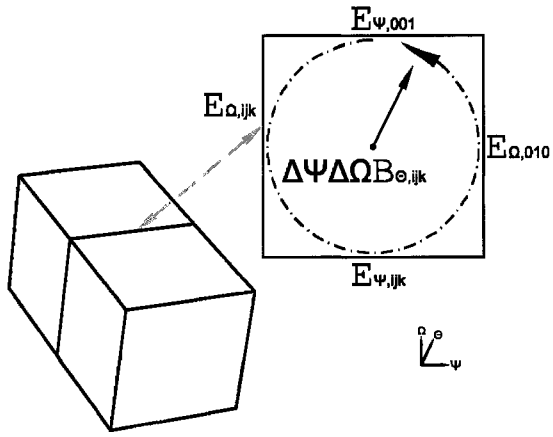


Figure 2-4: Adjacent computational cells with a shared face, shown in regular Cartesian coordinate system Θ, Ψ, Ω. Θ comes out of the page. Integer subscripts indicate addition to the base index ijk. **B** is located on the cell faces. **E** is located on the face edges.

the following expression for a single face

$$\left(\frac{\partial}{\partial t}B_{\Theta,ijk}\right)\Delta\Psi\Delta\Omega$$
$$= -\left((E_{\Psi,ijk})\Delta\Psi + (E_{\Omega,010})\Delta\Omega - (E_{\Psi,001})\Delta\Psi - (E_{\Omega,ijk})\Delta\Omega\right), \tag{2.32}$$

from which we arrive at the CT method of Evans and Hawley:

$$\frac{\partial}{\partial t}B_{\Theta,ijk} = -\left(\frac{E_{\Omega,010} - E_{\Omega,ijk}}{\Delta\Psi} - \frac{E_{\Psi,001} - E_{\Psi,ijk}}{\Delta\Omega}\right). \tag{2.33}$$

Here any reasonable approach may be used to estimate **E** without affecting the divergence-free property of CT. We will compute **E** on face edges using the second–order monotonic upwind approach originally described in Evans and Hawley (1988) and compute the divergence of the magnetic flux density on a computational cell with

$$\nabla \cdot \mathbf{B} \approx \frac{B_{\Theta,100} - B_{\Theta,ijk}}{\Delta\Theta} + \frac{B_{\Psi,010} - B_{\Psi,ijk}}{\Delta\Psi} + \frac{B_{\Omega,001} - B_{\Omega,ijk}}{\Delta\Omega}. \tag{2.34}$$

Note that the prescription of the magnetic flux density on cell faces, ensures that both the magnetic flux and divergence are evaluated correctly for the finite volume enclosed by the computational cell, while **E** on the cell edges ensures that the line intergals are correctly located on the surface of the finite volume enclosed by the computational cell. This type of consistency is required when constructing divergence free numerical methods (Toth, 2000).

2.6 Growth Rate of the Divergence of the Magnetic Flux Density

A desirable feature of the CT method is that if $\nabla \cdot \mathbf{B} = 0$ initially it will grow, in the worst case, linearly at a rate given by the floating point round off error introduced per time step. In order to determine a bound on the floating point round off error introduced we must look at the specific computations involved in the calculation of a single component of the curl of the electric field using equation (2.33). We have implemented the Van-Leer monotone upwind method of estimating the components of $\mathbf{E}$ which can be found in Evans and Hawley (1988). For a single component we have:

$$E_{\Omega,ijk} = \hat{v}_{\Psi,ijk}\hat{B}_{\Theta,ijk} - \hat{v}_{\Theta,ijk}\hat{B}_{\Psi,ijk}, \tag{2.35}$$

where the hatted components are located on cell edges and the three integer subscripts i, j, k identify the cell edge in question. We start by interpolating v_Ψ, and v_Θ from the cell center to the cell edge with:

$$\hat{v}_{\Psi,ijk} = \frac{1}{4}(v_{\Psi,ijk} + v_{\Psi,-100} + v_{\Psi,0-10} + v_{\Psi,-1-10}) \tag{2.36}$$

and

$$\hat{v}_{\Theta,ijk} = \frac{1}{4}(v_{\Theta,ijk} + v_{\Theta,-100} + v_{\Theta,0-10} + v_{\Theta,-1-10}). \tag{2.37}$$

This gives the velocity components needed in (2.35), now we need the magnetic field components. These are to be computed using a monotonic upwind interpolation. We first compute the forward and backward slopes:

$$S_{f\Theta,ijk} = \frac{B_{\Theta,ijk} - B_{\Theta,0-10}}{\Delta\Psi}, \qquad S_{b\Theta,ijk} = \frac{B_{\Theta,0-10} - B_{\Theta,0-20}}{\Delta\Psi}$$

and

$$S_{f\Psi,ijk} = \frac{B_{\Psi,ijk} - B_{\Psi,-100}}{\Delta\Omega}, \qquad S_{b\Psi,ijk} = \frac{B_{\Psi,-100} - B_{\Psi,-200}}{\Delta\Omega} \tag{2.38}$$

Followed by their products,

$$\Pi_{\Theta,ijk} = S_{b\Theta,ijk}S_{f\Theta,ijk} \tag{2.39}$$

and

$$\Pi_{\Psi,ijk} = S_{b\Psi,ijk}S_{f\Psi,ijk}. \tag{2.40}$$

(2.41)

The monotonic interpolation of B_Θ, and B_Ψ from face to edge can then be constructed by:

$$B^*_{\Theta,ijk} = \begin{cases} B_{\Theta,0-10} + \left(\frac{\delta_{b\Theta,ijk}\delta_{f\Theta,ijk}}{\delta_{b\Theta,ijk}+\delta_{f\Theta,ijk}}\right)(\Delta\Psi - \Delta t\hat{v}_{\Psi,ijk}), & \Pi_{\Theta,ijk} > 0 \\ B_{\Theta,0-10}, & \Pi_{\Theta,ijk} < 0 \end{cases}$$

and

$$B^*_{\Psi,ijk} = \begin{cases} B_{\Psi,-100} + \left(\frac{\delta_{b\Psi,ijk}\delta_{f\Psi,ijk}}{\delta_{b\Psi,ijk}+\delta_{f\Psi,ijk}}\right)(\Delta\Theta - \Delta t\hat{v}_{\Theta,ijk}), & \Pi_{\Psi,ijk} > 0 \\ B_{\Psi,-100}, & \Pi_{\Psi,ijk} < 0 \end{cases}$$

Finally an upwind selection gives the values of $\hat{B}_{\Theta,ijk}$ and $\hat{B}_{\Psi,ijk}$ which are needed in (2.35), namely

$$\hat{B}_{\Theta,ijk} = \begin{cases} B^*_{\Theta,ijk} & \hat{v}_{\Psi,ijk} > 0 \\ B^*_{\Theta,010} & \hat{v}_{\Psi,ijk} < 0 \end{cases} \tag{2.42}$$

and

$$\hat{B}_{\Psi,ijk} = \begin{cases} B^*_{\Psi,ijk}, & \hat{v}_{\Theta,ijk} > 0 \\ B^*_{\Psi,100}, & \hat{v}_{\Theta,ijk} < 0 \end{cases} \tag{2.43}$$

When attempting to estimate the rounding error accumulated during during the computation of (2.33), each individual floating point operation in this process must be accounted for, including the errors accumulated in the evaluation of (2.35), and all of the computations involved in forming its constituent components. First we need to know what kind of errors we can expect from floating point hardware. The IEEE 754 standard provides the answer. The rounding error introduced in individual floating point operations can be estimated using the following rules (Goldberg, 1991),

$$x \ominus y = (x - y)(1 + \delta_1), \tag{2.44}$$

$$x \oplus y = (x + y)(1 + \delta_2), \tag{2.45}$$

$$x \otimes y = x * y(1 + \delta_3), \tag{2.46}$$

where, $\oplus$, $\ominus$ and $\otimes$ indicate floating point operations, while $+$, $-$, and $*$ indicate their infinite precision counterparts; δ is a perturbation representing the upper bound on the rounding error introduced for a given operation. Specifically, $|\delta_1| \leq 2\epsilon$, $|\delta_2| \leq 2\epsilon$, and $|\delta_3| \leq \epsilon$ where ϵ is a bound on the rounding error made when rounding from infinite precision to a fixed

precision. A rule that accounts for the rounding errors accumulated during a summation over many terms is (Goldberg, 1991),

$$\sum_{j=1}^{n} \oplus x_j = \sum_{j=1}^{n} x_j(1+\delta_j) < (1+\delta_4) \sum_{j=1}^{n} x_j \tag{2.47}$$

where $|\delta_j| \leq (n-j)\epsilon$, and $\delta_4 = n\epsilon$. Machine epsilon where IEEE 754 double precision used is given by:

$$\epsilon = 2^{-53} \tag{2.48}$$

Given these rules, in order to find an upper bound on the rounding error involved in a specific series of floating point operations, we simply replace these operations with their infinite precision counterparts multiplied by the correct error factor, $(1+\delta)$, and simplify the resulting expressions. For convenience it may be assumed that the δ's are all positive. However, in practice, over a large number of operations they will be both positive and negative equally often in an unpredictable order. Thus, this assumption of positivity serves to over estimate the rounding error introduced.

For the interpolation of the components of velocity from cell center to cell edge we find an error factor of:

$$(1+\delta_3)(1+\delta_4), \tag{2.49}$$

while, in the case of the upwind interpolation of the components of the magnetic flux density from face center to cell edge we have in the worst case, an error factor of:

$$(1+\delta_1)^2(1+\delta_2)^3(1+\delta_3)^6(1+\delta_4). \tag{2.50}$$

Now we will compute a component of the electric field using (2.35), however we must take into account the errors accumulated in the interpolation of the components of both $\mathbf{v}$ and $\mathbf{B}$. In doing so we find an error factor of:

$$(1+\delta_1)^3(1+\delta_2)^3(1+\delta_3)^8(1+\delta_4)^2. \tag{2.51}$$

With the components of the electric field in hand we can compute their curl using (2.33). Again we must take into account the errors previously accumulated. In doing so we find an

error factor of:

$$(1+\delta_1)^5(1+\delta_2)^3(1+\delta_3)^9(1+\delta_4)^2. \tag{2.52}$$

Expanding this last expression in terms of ϵ we have (with many terms left out for brevity):

$$1+33\epsilon+508\epsilon^2+4852\epsilon^3+\cdots+4096\epsilon^{19}. \tag{2.53}$$

Since ϵ is a small number, the contributions made by the higher powers of ϵ are negligible. Dropping ϵ^2 and all higher power terms, we conclude that the order of rounding error per step, when using CT with the second–order monotonic upwind approach, is on the order of 10ϵ. Thus, given $\nabla \cdot \mathbf{B} = 0$ initially we can estimate the order of $\nabla \cdot \mathbf{B}$ at a later time by

$$O\left(\nabla \cdot \mathbf{B}|_{t=t_0+n\Delta t}\right) = 10n\epsilon O\left(max(|\mathbf{B}|)\right) \tag{2.54}$$

where n is the number of steps taken and the maximum should be taken over both space and time if $O(|\mathbf{B}|)$ changes over the time interval, and otherwise only spatially at $t = 0$. Given our assumption regarding the signs of the δ's in practice this will be a gross overestimate.

2.7 Semi-Conservative Numerics

An important aspect to consider when finite differencing the semi-conservative formulation of the ideal MHD system (see Section 1.4) is the difference scheme that will used to compute the current density,

$$\mathbf{j} = \nabla \times \mathbf{B}. \tag{2.55}$$

Notice from equations (1.28) - (1.32) that the current density has been pulled out of the conservative flux function, and as a result the conservative numerics discussed above are not applied to it. To be consistent with the momentum equation, the current density must be found on the cell, however where constrained transport methods are concerned the magnetic flux density is located on the cell faces. A simple approach to work through this issue is to first compute the components of the current density on the four parallel edges, then average the resulting values to the cell center. This is partially depicted in Figure 2-5. Here the face centered components of the magnetic flux density, shown in red, will be finite differenced to

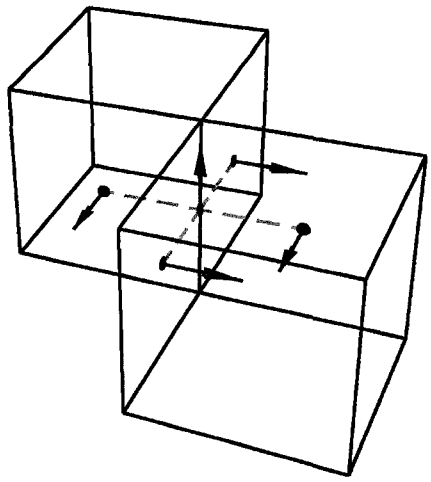


Figure 2-5: Stencil used to compute **j**.

attain a single component of the edge located current density, shown in blue. For instance, if the figure is assumed to show,

$$J_z = \frac{\partial B_y}{\partial x} - \frac{\partial B_x}{\partial y}, \tag{2.56}$$

then the two derivatives can be estimated using a first order two point formula like,

$$\frac{\partial B_x}{\partial y} \approx \frac{B_{x,ijk} - B_{x,i,j-1,k}}{\Delta y}. \tag{2.57}$$

It is possible to implement some other numerical scheme here. However, given the particular configuration of the components of the magnetic flux density, it will be somewhat tricky.

2.8 Pressure Positivity

At each finite-difference step we will calculate the pressure during both the estimation of the time step size and the computation of the numerical fluxes. For the conservation formulation of ideal MHD the pressure is computed with:

$$p = (\gamma - 1)\left[U - \frac{\rho v^2}{2} - \frac{B^2}{2}\right]. \tag{2.58}$$

In the case where either $v^2 \gg p$ or $B^2 \gg p$ then $U \approx \frac{B^2}{2}$ or $U \approx \frac{\rho v^2}{2}$ respectively. In either case one of the subtractions in (2.58), which is made using a finite precision, cancels

all but a few of least significant digits. Due to the accumulation of local truncation errors with each finite difference step the accuracy of the solution in the least significant digits, for all of the fields involved, is completely destroyed. After many steps these digits contain no information relating to the solution. The effect all of this has is that, in both cases, the computed value of pressure can be very bad, even negative. In reality, a negative pressure is an impossibility as it implies a negative temperature in the gas. This situation is called a catastrophic cancellation (Goldberg, 1991). When a catastrophic cancellation resulting in a negative pressure occurs the density soon after becomes negative, and the numerical solution is soon there after completely destroyed[1]. A number of approaches have been proposed to handle this situation. For instance one will solve another equation whereever it is detected that a catastrophic cancellation is likely to occur. For the case where the magnetic energy contributes most of the total energy (Balsara and Spicer, 1999) has developed an approach, and for the case where the kinetic energy constributes most of the total energy (Ryu et al., 1993) has developed an approach. For the case where the magnetic energy constributes most of the total energy, an alternative approach makes use of the semi-conservative form of the energy equation (Raeder, 2003). If the semi-conservative formulation of the ideal MHD equations are used the pressure is computed according to:

$$p = (\gamma - 1)\left[e_T - \frac{1}{2}\rho \mathbf{v} \cdot \mathbf{v}\right]. \tag{2.59}$$

Here the magnetic energy does not enter the calculation, hence there is no possibility of a catastrophic cancellation.

[1]The solution will be destroyed if a hardware exception is not thrown, because computing the sound speed using a negative density results in taking the square route of a negative number which is an invalid operation on most floating point hardware. Activating floating point exceptions proves very useful when debugging an MHD code as the time and location of otherwise illusive events are pinpointed precisely.

CHAPTER 3

BLOCK STRUCTURED ADAPTIVE MESH REFINEMENT

3.1 Chapter Abstract

In this chapter a review of block structured adaptive mesh numerical methods for hyperbolic systems is presented. I will discuss the numerical challenges which the introduction of arbitrary resolution changes in Cartesian grids bring about. I will present a historical development of solutions to these issues and follow it up with a presention of a new numerical method that I have developed with Joachim Raeder which extends MHD Constrained Transport numerical methods for use on block structured adaptive grids. In addition to presenting this new research, I will present my research on visualization of scientific data produced by block structured adaptrive mesh numerical codes.

3.2 Introduction and Terminology

Adaptive mesh refinement addresses the following problem: Given a fixed amount of computing power how can one get the most accurate solution to a set of given PDE's? Accuracy of the solution depends on the local truncation error of the finite-difference method, which is a function of $\Delta\mathbf{x}$ and Δt. One way to get better results is to decrease the grid spacing $\Delta\mathbf{x}$. However, each decrease in $\Delta\mathbf{x}$ for a three-dimensional problem drastically increases the amount of computation required to advance the solution in time. The solution must be found on more computational cells and for advection type problems the CFL condition dictates that more time steps must be taken to reach the same end time. Adaptive mesh

refinement addresses the problem optimally in the sense that $\Delta \mathbf{x}$ is decreased only in specific areas leaving $\Delta \mathbf{x}$ larger elsewhere, targetting computational resources to where they will make the most difference with the effect that a higher accuracy solution than would have been otherwise possible is obtained for a given amount of computational effort.

Block structured adaptive mesh refinement (AMR) was developed in a series of papers by M.J.Berger et al., (Berger and Oliger, 1984), (Berger, 1987), (Berger and Colella, 1989), (Bell et al., 1994). In AMR, as in many other finite difference procedures for solving time dependent hyperbolic systems, the solution is to be found on a box shaped area which is called the probelm domain. The solution will be eveloved in time by the repetition of many small time steps until the desired end time is achieved. The following points differentiate block structured AMR from other finite difference solution procedures developed for use on unifrom grids:

1. The solution has a higher resolution in specific spatial locations which may evolve in time. Resolution changes are limited to an integer increment, called the refinement ratio. The computational grids are refined in both space and time due to the coupling of the spatial grid to the temporal grid via the CFL condition.

2. Areas are composed of computational cells having the same resolution and are treated together under the abstraction of a Level. A Level is a collection of cell-wise disjoint[1] box shaped computational grids.

3. A base level, often identified as Level 0, is always defined and by convention fills the entire problem domain.

4. The refined levels which have a higher computational resolution than the base level, are introduced as needed. These are given integer identifiers from 1 to M, where level M is the finest level and the value M can increase or decrease as needed such that $0 \leq M \leq M_{max}$, where M_{max} is the maximum level. Refined levels unlike the base

[1]In the following discusion we need to define some terms and conventions that allow precise specification of certain groups of computational cells. These terms and a few others relating to AMR are given in the glossary which has been included in Appendix 5.7

level need not fill the entire problem domain but must be decomposed into a set of cell-wise disjoint boxes.

5. All refined levels must be nested with respect to their parent level such that there is at least one level i cell between level i-1 and level i+1, except at the physical boundary. However, the nesting of levels places no restriction on the arrangement of boxes that compose them and the relationship between the boxes of one level and the boxes of another.

6. On nested Levels boundary conditions will be provided with the problem specification for computational cells which are out side of the problem domain, and via a space-time interpolation for computational cells which are on the interior of the problem domain. The interior boundary conditions are called coarse–fine boundary conditions. Application of the coarse–fine boundary conditions requires that the solution is found bottom to top, starting first with Level 0 and proceeding to Level M.

7. When a refined level is introduced or expands the newly refined computational cells must be initialized. The initialization is accomplished via an interpolation (also known as a prolongation) from the next coarser level. For this reason it is important that coarse solution maintain the most accurate representation possible, and not get too far out of sync with the refined solution. To this end, when the solutions on refined grids become synchronous with the solution on next coarser level the refined solution is injected (also called a restriction). The restriction will be handled by the coarser level, and proceeds from the top of the grid hierarchy to the bottom, starting with Level M-1.

The above characteristics were presented in Berger and Oliger (1984), where rotated rectilinear grids were used. The development of the special case of coordinate axis aligned Cartesian grids which are used in my code can be found in Berger and Colella (1989).

Initially a grid generation procedure is applied to the base level an arbitrary number of times creating a series of successively refined, properly nested levels. The process of

identifying the regions of the problem domain which will have a higher resolutions is called tagging. Tagging is often based on an error estimate computed in the process of solving the system of differential equations (Berger and Oliger, 1984) or it can be a problem specific condition such as the magnitude of a scalar exceeding a threshhold value. A side effect of tagging cell by cell is that cells marked for tagging are always fully refined. The grid generation procedure takes a set of tags, which are simply an array having the same dimensions as the computational grid filled with ones and zeros, for each level in a level hierarchy, and produces a new box layout for each of the levels from which a new hierarchy of computational grids will be generated. The entire process of taking an existing solution on a hierarchy of computational grids, identifying regions to refine and generating a new hierarchy of computational grids, and initializing the new grid hierarchy with the existing solution is known collectively as regridding.

Block structured AMR provides a great deal of flexibility, because any tagging criteria may be used. Further, different levels of refinement may employ different numerical methods or even solve different systems of equations. For example shocks in a gas have a viscous profile with a thickness on the order of a few mean free path lengths, hence one could solve the Naiver Stokes equations locally around the shock on computational grid resolution on the order of the mean free path, while elsewhere solving the Euler equations on a much larger computational grid resolution, applying the relevent physics on the appropriate scale.

The process of evolving the solution in time is called the run. The collection of all levels in play at a given time during a run is called the level or grid hierarchy. As the run proceeds, the regridding procedure periodically takes the existing hierarchy of grids and generates a new hierarchy of grids perhaps with more levels, perhaps with fewer, based on a set of tags, one for each level in the hierarchy. It is important to recognize that regridding a level i affects the shape of all of the levels from level i+1 to the finest level, level M. Due to the fact that each refined grid must be initialized from its parent, the integer, M, which identifies the finest level, may only increase by one in each regridding. However M is free to decrease to 0 in a single regridding step, if no cells on Level 0 are tagged for refinement.

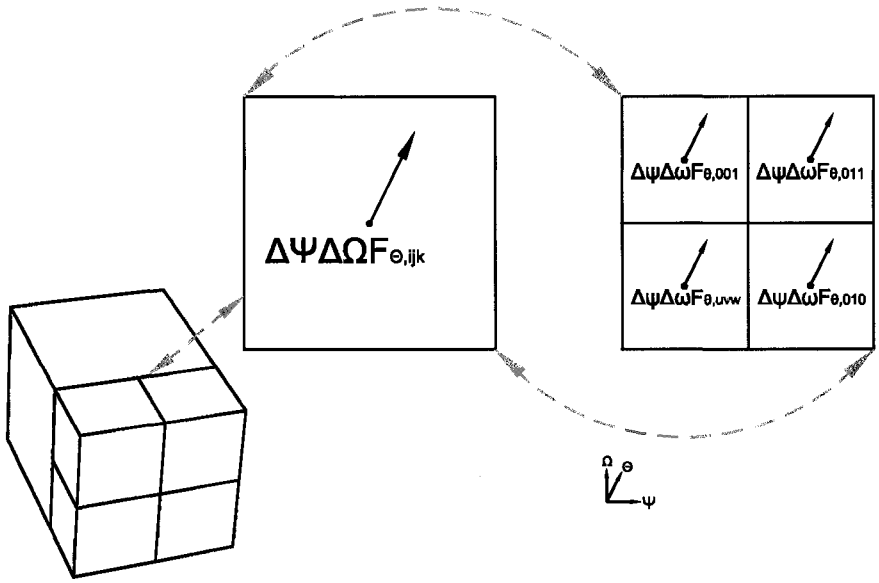


Figure 3-1: Flux conservation across a coarse–fine interface. Four fine cells share a face with a coarse cell at the coarse fine grid interface, shown in regular Cartesian coordinate system Θ, Ψ, Ω where Θ comes out of the page. Integer subscripts indicate addition to the base index ijk (coarse grid) or uvw (fine grid). Components of the flux are located on the cell faces.

3.3 Flux Conservation at the Coarse-Fine Grid Interface

A primary concern when using AMR techniques to solve hyperbolic systems is numerical conservation. The numerical conservation conditions in equation (2.11) require that the flux across the faces shared by coarse and fine cells on the coarse–fine grid interface is equivalent in both grids, namely that,

$$\Delta\Psi\Delta\Omega\hat{f}^{n}_{\Theta,i,j,k} = \Delta\psi\Delta\omega\sum_{q=1}^{r}\left(\hat{f}^{m+q}_{\theta,u,v,w} + \hat{f}^{m+q}_{\theta,010} + \hat{f}^{m+q}_{\theta,001} + \hat{f}^{m+q}_{\theta,011}\right). \tag{3.1}$$

where r is the refinement ratio. However, this is not true a priori due to the difference in LTE between the coarse and fine solution. As a result conservation is immediately violated after a restriction is applied on computational cells which abut the refined region. Specifically, the conditions specified in (2.11) are not satisfied when the numerical flux is evaluated in the cells across which the resolution increases. For example numerical conservation will not be satisfied in the Level 0 cells in Figure 3-3.a which have a face coincident with the coarse–fine interface (dashed blue line). The situation is depicted in greater detail in Figure

3-1. A Cartesian coordinate system in the orthogonal basis vectors $\hat{\Theta}$,$\hat{\Psi}$,$\hat{\Omega}$ is introduced. Discretized quantities are identified by the subscripts ijk. These subscripts when used with the numerical fluxes $\hat{\mathbf{f}}$, identify the base face location. Integer subscripts give the relative location with respect to the base face; for example, $010 \equiv i, j{+}1, k$. Given the requirement of a bottom to top solution approach dictated by the interior coarse–fine boundary conditions, a natural way to satisfy (3.1) is to construct and apply a correction to the solution after a restriction has been applied from the next finer level. Due to the coupling between spatial and temporal discretization provided by the CFL condition the next finer level will be synchronous after the it has advanced r steps, where r is the grid refinement ratio. The correction that will restore flux conservation to the solution can be constructed as follows (Berger and Colella, 1989):

$$\delta F^n_{\Theta,i,j,k} = \hat{f}^n_{\Theta,i,j,k} - \frac{\Delta\psi\Delta\omega}{\Delta\Psi\Delta\Omega}\sum_{q=1}^{r}\left(\hat{f}^{m+q}_{\theta,u,v,w} + \hat{f}^{m+q}_{\theta,010} + \hat{f}^{m+q}_{\theta,001} + \hat{f}^{m+q}_{\theta,011}\right). \tag{3.2}$$

Here the cumulative difference between the flux on this level and the next finer level is computed. The correction takes the form of a finite difference step which applies the difference in fluxes to coarse cells that abut the resolution change

$$\mathbf{u}^{n+1}_{i,j,k} = \mathbf{u}^{n+1}_{i,j,k} - \left(\frac{\Delta t}{\Delta\Theta}\right)\delta F^n_{\Theta,i,j,k}. \tag{3.3}$$

The process of accumulating the flux difference using equation (3.2) and applying a correction given by equation (3.3) solution such that equation (3.1) is satisfied is called refluxing.

The magnetic flux density is a fundamentally different type of conserved quantity due to its association with with the divergence free constraint. Applying the divergence theorem to (2.30) we see that

$$\Phi = \iint \mathbf{B}\cdot d\vec{\mathbf{S}} = \iiint (\nabla\cdot\mathbf{B})\, dV = 0. \tag{3.4}$$

This equation explicitly illustrates the connection between magnetic flux conservation and the divergence-free condition on a computational cell. Note that after a restriction operation takes place, if flux has not been conserved across the coarse–fine grid interface, the divergence-free property of the coarse solution will be destroyed in cells that abut the refined region. This occurs in those cells due to the fact that the cell is divergence-free prior to the

restriction, and that the restriction operation modifies the value of a single component of the magnetic flux density on the face coincident with the coarse–fine grid interface, without changing the components of the magnetic flux density on the other faces of that cell. In light of equation (3.4) the non-zero divergence in these cells implies a loss of magnetic flux conservation at the coarse–fine grid interface. Thus, a non-zero divergence appearing in coarse cells which abut the coarse–fine grid interface indicates that magnetic flux has not been conserved across the coarse–fine grid interface. Where constrained transport (CT) numerical schemes are applied care must be taken with respect to restriction, prolongation and magnetic flux conservation across the coarse–fine grid interface. As such a treatment which differs from the refluxing treatment of the hydrodynamic quantities must be applied. An approach analogous to the refluxing of hydrodynamic quantities can be developed (Balsara, 2001). This results in a procedure known as curl-refluxing, where the differences in the electric field $\mathbf{E}$ at each step between this level and the next finer level are accumulated, and a correction to the magnetic flux density is applied by taking the curl of the accumulated differences. I will present an alternative approach which will conserve magnetic flux without a refluxing step.

3.4 Magnetic Flux Conservation at the Coarse-Fine Grid Interface

In order to use the CT method with AMR grids, we must address the issue of magnetic flux conservation across the coarse–fine grid interface. Rather than develop a correction or reflux step we will set out to design a procedure for a finite-difference scheme which will automatically conserve magnetic flux and preserve divergence in all of the grids involved. The situation is depicted in Figure 3-2. Θ, Ψ, Ω, ijk, and $\Delta\Theta$, $\Delta\Psi$, $\Delta\Omega$ are used to identify coordinate directions, base index, and computational cell edge lengths, respectively, on the coarse grid, while θ, ψ, ω, uvw, and $\Delta\theta$, $\Delta\psi$, $\Delta\omega$ are used for the same purpose on the fine grid. We introduce superscripts n and m to denote discrete temporal location on the coarse

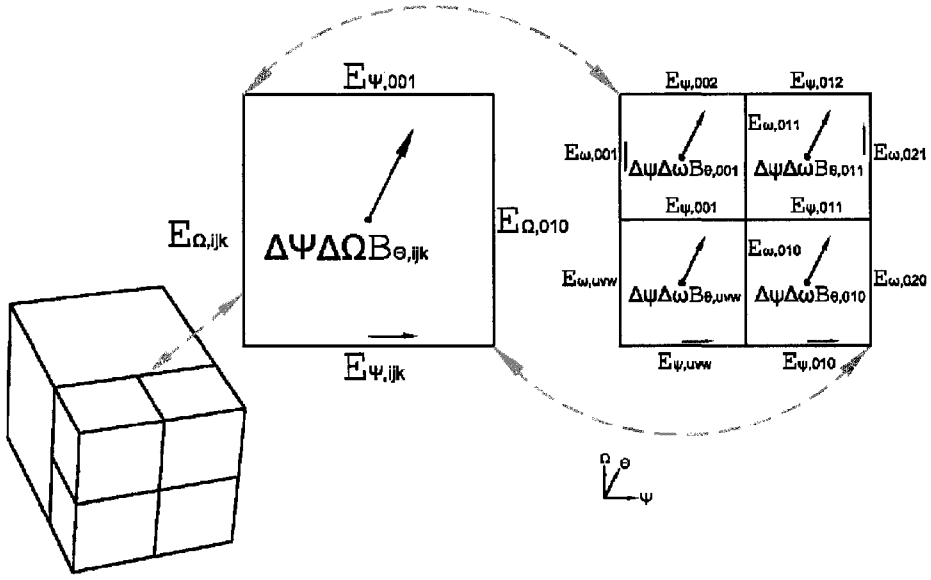


Figure 3-2: Coarse–fine grid interface. Four fine cells share a face with a coarse cell at the coarse fine grid interface, shown in regular Cartesian coordinate system Θ, Ψ, Ω where Θ comes out of the page.

and fine grids respectively and we use a grid refinement ratio of 2. The conservation of magnetic flux across the coarse–fine grid interface implies that both

$$\Delta\Psi\Delta\Omega B^n_{\Theta,ijk} = \Delta\psi\Delta\omega \left(B^m_{\theta,uvw} + B^m_{\theta,010} + B^m_{\theta,001} + B^m_{\theta,011} \right) \tag{3.5}$$

and

$$\Delta\Psi\Delta\Omega B^{n+1}_{\Theta,ijk} = \Delta\psi\Delta\omega \left(B^{m+2}_{\theta,uvw} + B^{m+2}_{\theta,010} + B^{m+2}_{\theta,001} + B^{m+2}_{\theta,011} \right) \tag{3.6}$$

are satisfied. Given a grid refinement of 2, if the coarse and fine solutions are synchronous at a time specified by m and n, they will become synchronous again at the time specified by $n+1$ and $m+2$, due to the CFL restriction on the refined grid. We seek to construct a numerical scheme so that given (3.5), (3.6) is satisfied automatically. In other words we would like the time rate of change of the magnetic flux during the advance on the shared coarse–fine faces to be equivalent, specifically:

$$\begin{aligned} &\Delta\Psi\Delta\Omega \frac{\partial}{\partial t}(B_{\Theta,ijk}) \\ &= \Delta\psi\Delta\omega \left(\frac{\partial}{\partial t}(B_{\theta,uvw}) + \frac{\partial}{\partial t}(B_{\theta,010}) + \frac{\partial}{\partial t}(B_{\theta,001}) + \frac{\partial}{\partial t}(B_{\theta,011}) \right) \end{aligned} \tag{3.7}$$

Here we must take into account that we evaluate the right side of (3.7) in 2 steps due to the CFL restriction, while the left side of (3.7) is obtained in a single step. Examining the lower-left face of the fine level we find that the magnetic field after two CT time steps is given by the expression:

$$\begin{aligned} B^{m+2}_{\theta,uvw} &= B^{m}_{\theta,uvw} \\ &- \frac{\Delta t}{\Delta\psi\Delta\omega}\Big[(E^{m}_{\psi,uvw}+E^{m+1}_{\psi,uvw})\Delta\psi+(E^{m}_{\omega,010}+E^{m+1}_{\omega,010})\Delta\omega \\ &\qquad -(E^{m}_{\psi,001}+E^{m+1}_{\psi,001})\Delta\psi-(E^{m}_{\omega,uvw}+E^{m+1}_{\omega,uvw})\Delta\omega\Big]. \end{aligned} \tag{3.8}$$

Therefore by setting:

$$\left.\begin{aligned} E^{m}_{\psi,uvw} &= E^{m+1}_{\psi,uvw} \\ E^{m}_{\omega,010} &= E^{m+1}_{\omega,010} \\ E^{m}_{\psi,001} &= E^{m+1}_{\psi,001} \\ E^{m}_{\omega,uvw} &= E^{m+1}_{\omega,uvw} \end{aligned}\right\}, \tag{3.9}$$

we obtain the desired result, a discretized expression for time change of magnetic flux on a particular fine level face over the time spanned by the coincident coarse face's time step, namely,

$$\begin{aligned} &\Delta\psi\Delta\omega\left(\frac{B^{m+2}_{\theta,uvw}-B^{m}_{\theta,uvw}}{2\Delta t}\right) \\ &\quad = -(E^{m}_{\psi,uvw}\Delta\psi+E^{m}_{\omega,010}\Delta\omega-E^{m}_{\psi,001}\Delta\psi-E^{m}_{\omega,uvw}\Delta\omega) \end{aligned} \tag{3.10}$$

Substituting (2.32) for the left side of (3.7) and (3.10) and expressions like it for the other three fine faces into (3.7), we obtain

$$\begin{aligned} &E^{n}_{\Psi,ijk}\Delta\Psi+E^{n}_{\Omega,010}\Delta\Omega-E^{n}_{\Psi,001}\Delta\Psi-E^{n}_{\Omega,ijk}\Delta\Omega \\ &\quad = (E^{m}_{\psi,uvw}+E^{m}_{\psi,010})\Delta\psi+(E^{m}_{\omega,020}+E^{m}_{\omega,021})\Delta\omega \\ &\qquad -(E^{m}_{\psi,002}+E^{m}_{\psi,012})\Delta\psi-(E^{m}_{\omega,uvw}+E^{m}_{\omega,001})\Delta\omega. \end{aligned} \tag{3.11}$$

We can then equate coarse–fine terms which are coincident to arrive at the following set of conditions:

$$E^{n}_{\Psi,ijk}\Delta\Psi = (E^{m}_{\psi,uvw}+E^{m}_{\psi,010})\Delta\psi \tag{3.12}$$

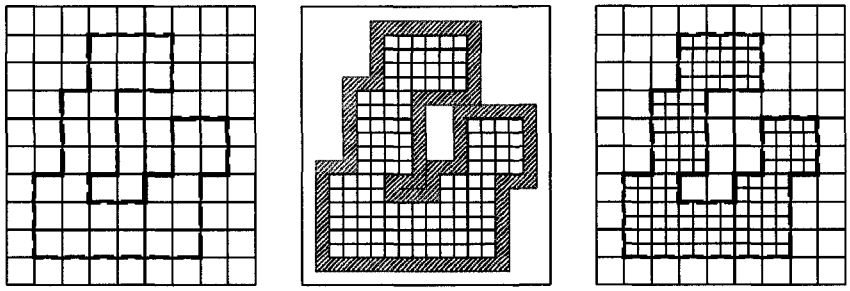


Figure 3-3: Example grid hierarchy. (a) base level, (b) refined level , (c) visualization grid (c). The coarse–fine interface is shown (thick broken line). The refined grid is shown with its ghost cells(hatched areas).

$$E^n_{\Psi,001}\Delta\Psi = (E^m_{\psi,002} + E^m_{\psi,012})\Delta\psi \tag{3.13}$$

$$E^n_{\Omega,010}\Delta\Omega = (E^m_{\omega,020} + E^m_{\omega,021})\Delta\omega \tag{3.14}$$

$$E^n_{\Omega,ijk}\Delta\Omega = (E^m_{\omega,uvw} + E^m_{\omega,001})\Delta\omega \tag{3.15}$$

Equations (3.12)–(3.15) can be satisfied by setting:

$$\left.\begin{aligned} E^m_{\psi,uvw} &= E^m_{\psi,010} = E^n_{\Psi,ijk} \\ E^m_{\psi,002} &= E^m_{\psi,012} = E^n_{\Psi,001} \\ E^m_{\omega,020} &= E^m_{\omega,021} = E^n_{\Omega,010} \\ E^m_{\omega,uvw} &= E^m_{\omega,001} = E^n_{\Omega,ijk} \end{aligned}\right\}. \tag{3.16}$$

Thus we have the set of conditions, namely (3.9) and (3.16), which taken together guarantee that if we start with the flux conserved on cells which share faces across the coarse–fine grid interface, we will maintain flux conservation automatically during the time update. These are in essence a set of boundary conditions for the coarse–fine grid interface which we apply at each fine time step. In doing so we enforce the conservation of magnetic flux on the grid interface.

Before advancing the solution, we apply the conditions, (3.9) and (3.16), on the coarse–fine boundary insuring flux conservation during the advance. We treat the edges of the fine cell faces that are coincident with the coarse–fine grid interface as belonging to the ghost cells, interpolating from the coarse level there. This is reasonable because these edges are infact part of the fine level ghost cells. A constant interpolation is reasonable for these edges

because the edges of the fine cells on the coarse–fine interface are infact spatially coincident with the coarse cell edges that are used as the source of the interpolation. The situation is depicted in Figure 3-3, where coarse–fine interface is shown on the left, and the fine grid is shown on the right with its ghost cells. It is expected that the use of these conditions leads to similar results compared to the curl-refluxing approach. For both approaches flux across the coarse–fine interface, as calculated on the fine level, is calculated using information contained in the fine level ghost cells, which, except at the physical boundary of the problem space, contain at best coarse level accurate information. We posit that in most situations the affects on the solution due to using our approach are minimal compared to the traditional approach, however this remains to be shown. A primary concern in this regard is that of convergence. We must show that the use of the proposed coarse–fine boundary conditions does not reduce the rate of convergence to below second order. A convergence study has yet to be completed.

3.5 Divergence Preserving Restriction and Prolongation Operations

A key property of the CT method for solving Faraday's law is that equation (1.34) is satisfied to within roundoff error. Thus, a crucial aspect of AMR where CT numerics are concerned is that inter-level operations must not introduce divergence. There are two possibilities for this to occur. First, when coarse and fine grid solution become synchronous, where the coarse grid is covered by refined regions, the coarse grid is updated by a fine-to-coarse averaging procedure known as a restriction. Second, as the solution evolves and grids expand to cover previously unrefined areas, newly created refined grids are initialized via an interpolation from the coarse grid known as a prolongation. Both restriction and prolongation operations applied to the magnetic flux density must be divergence-preserving if the solution is to have the divergence-free property that CT promises. A number of divergence preserving prolongation/interpolation approaches have been developed, for example

Mackay et al. (2006), Handscom (1991), Balsara (2001), and, Toth and Roe (2002). I have chosen to use the divergence-preserving restriction and prolongation formulas described in Toth and Roe (2002), primarily for convenience as these are presented in a clear and succinct manner. For reference the formulas developed these are presented here. Interpolation to fine cell faces, coincident with 1 of 6 coarse cell faces is given by:

$$\hat{\Phi}_{x,\pm 2,j,k} = \frac{1}{4}\left(\Phi_{X,\pm 2,0,0} + j\frac{\partial}{\partial y}\Phi_{X,\pm 2,0,0} + k\frac{\partial}{\partial z}\Phi_{X,\pm 2,0,0}\right) \tag{3.17}$$

$$\hat{\Phi}_{y,i,\pm 2,k} = \frac{1}{4}\left(\Phi_{Y,0,\pm 2,0} + i\frac{\partial}{\partial x}\Phi_{Y,0,\pm 2,0} + k\frac{\partial}{\partial z}\Phi_{Y,\pm 2,0,0}\right) \tag{3.18}$$

$$\hat{\Phi}_{z,i,j,\pm 2} = \frac{1}{4}\left(\Phi_{Z,0,0,\pm 2} + i\frac{\partial}{\partial x}\Phi_{Z,0,0,\pm 2} + j\frac{\partial}{\partial y}\Phi_{Z,\pm 2,0,0}\right), \tag{3.19}$$

where the notation used follows that used in Toth and Roe (2002). A coordinate system has been introduced such that the origin is located at the cell center of the coarse computational cell. The six coarse cell face centers are located at $\pm 2, 0, 0$, $0, \pm 2, 0$, and $0, 0, \pm 2$. In the formulas a subscript written like $i, j, k = \pm 1$ is intended to indicate all of the permutations which can be obtained by successively substituting ± 1 into each index. Hatted components are from the refined cell. The interpolations to fine cell faces on the interior of the coarse cell are given by:

$$\hat{\Phi}_{x,0,j,k} = \frac{1}{2}(\hat{\Phi}_{x,2,j,k} + \hat{\Phi}_{x,-2,j,k}) + \frac{\partial^2 \Phi_x}{\partial x^2} + k(\Delta z)^2 \frac{\partial^3 \Phi_y}{\partial x \partial y \partial z} + j(\Delta y)^2 \frac{\partial^3 \Phi_z}{\partial x \partial y \partial z} \tag{3.20}$$

$$\hat{\Phi}_{y,i,0,k} = \frac{1}{2}(\hat{\Phi}_{y,i,2,k} + \hat{\Phi}_{y,i,-2,k}) + \frac{\partial^2 \Phi_y}{\partial y^2} + k(\Delta z)^2 \frac{\partial^3 \Phi_x}{\partial x \partial y \partial z} + i(\Delta x)^2 \frac{\partial^3 \Phi_z}{\partial x \partial y \partial z} \tag{3.21}$$

$$\hat{\Phi}_{z,i,j,0} = \frac{1}{2}(\hat{\Phi}_{z,i,j,2} + \hat{\Phi}_{z,i,j,-2}) + \frac{\partial^2 \Phi_z}{\partial z^2} + i(\Delta x)^2 \frac{\partial^3 \Phi_y}{\partial x \partial y \partial z} + j(\Delta y)^2 \frac{\partial^3 \Phi_x}{\partial x \partial y \partial z} \tag{3.22}$$

The situation is depicted in Figure (3-4). The construction of these formulas is such that flux is conserved on the 6 coarse cell faces and the 24 coincident fine cell faces and each fine cell divergence equals its parent coarse cell divergence. There are 36 fine cell faces while this construction only provides 31 equations. To complete the system Toth and Roe specify that

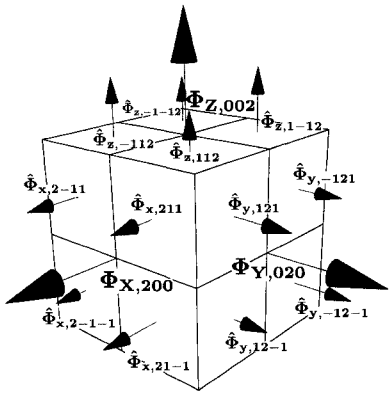


Figure 3-4: Divergence preserving interpolation stencil. Field components of the coarse cell (blue), and eight fine cells (red).

the six curls of the fine solution around the six fine cell edges starting from the origin have the same value as the curl estimated from the coarse solution. The resulting formulas are second order accurate. Often one needs to interpolate into a region adjacent where existing data must be preserved. In that case one simply uses existing data in the formulas, and the divergence is preserved.

The restriction operation is given by flux conservation i.e.:

$$\Phi_{X,\pm 2,0,0} = \sum_{j,k=\pm 1} \hat{\Phi}_{x,\pm 2,j,k} \tag{3.23}$$

with similar formulas for y and z components. On a coarse cell after a restriction we have a divergence of:

$$\nabla \cdot \mathbf{B}_{0,0,0} = \frac{a}{A}\left(\frac{\partial}{\partial x}\sum_{j,k=\pm 1}\left(\hat{B}_{x,\pm 2,j,k} + \hat{B}_{x,0,j,k}\right)\right.$$
$$\left. + \frac{\partial}{\partial y}\sum_{i,k=\pm 1}\left(\hat{B}_{y,i,\pm 2,k} + \hat{B}_{y,i,0,k}\right) + \frac{\partial}{\partial z}\sum_{i,j=\pm 1}\left(\hat{B}_{z,i,j,\pm 2} + \hat{B}_{z,i,j,0}\right)\right)$$

where a is the area of a refined cell face, A is the area of the coarse cell face, and hatted components are the values on the fine cell. Rearranging we obtain,

$$\nabla \cdot \mathbf{B}_{ijk} = \frac{a}{A}\left(\sum_{i,j,k=\pm\frac{1}{2}} \nabla \cdot \hat{\mathbf{B}}_{ijk}\right). \tag{3.24}$$

Thus, after a restriction the divergence on the coarse grid will be the average of that on the fine grid.

During the application of the coarse–fine boundary conditions a time interpolation is needed. The time interpolation to time α where $0 \leq \alpha \leq 1$ is given by:

$$\mathbf{B}^{\alpha} = \alpha \mathbf{B}^{1} + (1 - \alpha)\mathbf{B}^{0} \tag{3.25}$$

The divergence of the time interpolated solution is given by:

$$\nabla \cdot \mathbf{B}^{\alpha} = \alpha \nabla \cdot \mathbf{B}^{1} + (1 - \alpha)\nabla \cdot \mathbf{B}^{0}. \tag{3.26}$$

Thus the divergence is bound between the values of the two solutions. It is important to note that all of these formulas are divergence preserving, as opposed to divergence free or divergence cleaning, in that what ever divergence is present before the operation in question will be carried over in the result.

3.6 Visualization of AMR Data

When AMR is used to solve a system of differential equations the solution is a collection of levels. For visualization and analysis solution data having the highest available resolution in the hierarchy is needed. However, all levels have the solution on their entirety even where they have been further refined. Consequently there is a hierarchy of redundant less accurate solutions. Although this less accurate data is an integral part of the solution procedure, once the run has completed it is typically not needed. In most cases the less accurate solution should not be used for visualization and analysis. The process of removing all but the most highly resolved solution data is called flattening or cleaning the data set. Once the solution hierarchy has been flattened visualization and analysis can be performed.

There are many issues that come up in the post-processing[2] of AMR data due to the fact that standard numerical and visualization algorithms designed for data on regular grids are not designed to handle the changes in grid resolution present in AMR data. An additional complication is that many post-processing algorithms require nodal data rather than the cell constant data generated by a finite-difference solver. If the AMR data is not treated

[2]A term which will be used in place of "visualization and analysis".

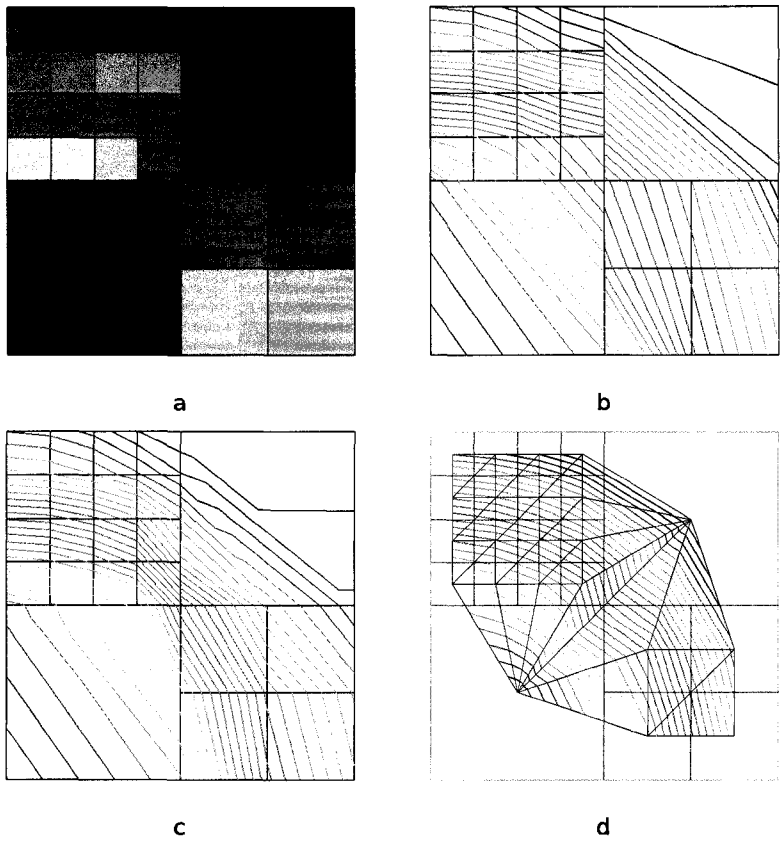


Figure 3-5: AMR visualization issues and approaches. a) A cleaned three level AMR grid, ready for analysis and visualization b) Data visualization using a pixel/voxel type cell results in broken contours c) Data visualization using a polygonal cell is correct d) Data visualization, by introducing nodes at cell centers and triangulating/tetrahedralizing the result is correct

with care issues in the form of artifacts or breaks will arise at coarse–fine grid interface. For example, interpolation, a core component in many post-processing algorithms, such as contour generation, slicing, and stream line generation, must be AMR aware to produce correct results. Typically an interpolation routine will interpolate between the eight nodes that surround a three dimensional computational cell to provide an estimation of a scalar or vector at a location on the interior of the cell. The result of the interpolation is often used in further analysis or to construct some element of a visualization. However, on three dimensional computational cells which abut an increase in grid resolution there are more than eight nodes to consider. If the interpolation algorithm does not use all of the nodes available, errors are introduced.

The nature of these errors can easily be demonstrated using a two dimensional example as the issues are analogous in three dimensions. The visualizations presented were made using the VTK imaging tool kit[3][4]. The situation is depicted in Figure 3-5. A hypothetical 2D AMR data set is shown with 3 levels of refinement. In all of the sub-figures the same data set has been used, where a scalar field initialized to $s = cos(x)cos(y)$ with $0 \leq x \leq 1$ and $0 \leq y \leq 1$. Figure 3-5.a shows the unprocessed data, where the solution is constant on each computational cell. Unfortunately about the only thing one can do with cell constant data in VTK is to color a visualization cell accordingly. For a three dimensional data set composed of cells with cell constant data the situation is unacceptable since one needs to sample the data set in order to see inside it and most sampling operations rely on the presence of nodal data. To enable further analysis one will have to convert the cell constant data to nodal data. Figure 3-5.b shows a contour plot of the data after the cell data has been converted to node data. A pixel type cell, which is limited to four nodes, from the VTK toolkit has been used. In three dimensions the analogous cell type is called a voxel and has eight nodes. When either of these cell types are used broken contour lines are produced since the interpolator in the coarse cell that abuts the resolution increase does not use all of the which are coincident with its surface. Nodes which are ignored are called dangling nodes. Figure 3-5.c shows the same visualization except the cell type has been changed to a polygonal cell which supports an arbitrary number of nodes. As a result all of the available nodes are used in the interpolation and the contour lines are not broken. Unfortunately, at the time that this project was started there was no three dimensional equivalent to the polygonal cell in VTK. However, in the latest release there is such a cell. It is called a convex point set cell. The convex point set cell can be constructed with an arbitrary number of nodes, and thus is inherently AMR aware. Finally, Figure 3-5.d shows

[3]hhtp://www.vtk.org

[4]Although this discussion pertains specifically to VTK, it should be noted that currently the two applications available for visualizing CHOMBO AMR data, CHOMBO Vis (http://seesar.lbl.gov/anag/chombo/chombovis.html), and Visit (https://wci.llnl.gov/codes/visit/) are simply GUI's wrapping VTK, thus VTK issues are carried over to both CHOMBO Vis and Visit.

an alternate approach where, rather than convert the cell data to nodal data, nodes are introduced at cell centers and the resulting point set is tetrahedralized. This approach suffers none of the aforementioned trouble in the regions of resolution changes. I have used the tetrahedralization approach to produce the figures presented. Unfortunately the tetrahedralization is far too slow for use on all but the smallest 3D data sets. In future work the latest developments in VTK will have to be leveraged to make post processing both accurate and practical.

CHAPTER 4

CODE DESIGN AND IMPLEMENTATION DETAILS

4.1 Chapter Abstract

In this chapter I present the CHOMBO AMR C++ package and how I have extended and modified it to implement the new CT MHD solver that I derived in the preceding chapters. I will present the details of my investigation of the adaptive grid generation, and hierarchical time-stepping algorithms, specific to the CHOMBO implementation. This investigation lead to the discovery of a bug in the CHOMBO implementation which could lead to a numerical violation of the CFL condition and as a consequence the introduction of numerical instability. I will present how I have modified CHOMBO to deal with this issue.

4.2 CHOMBO

The CHOMBO package[1] is a set of C++ classes which provide much of the infrastructure needed for a block-structured AMR code. CHOMBO is currently under development, and contains literally 100's of C++ classes which provide a wide variety of functionality. Support is included for Parabolic, Hyperbolic and Elliptic type problems. The Berger-Rigoutsos algorithm for mesh generation is included, along with object oriented data types for Levels, layouts, and computational boxes. Many of the data types have been implemented with parallel applications in mind and have features which facilitate their use on MPI based computer clusters. Much of the CHOMBO's infrastructure is accessed via C++ polymor-

[1] http://seesar.lbl.gov/ANAG/chombo/

phism where CHOMBO provides an interface definition from which we derive a problem specific implementation from. For hyperbolic problems CHOMBO provides the AMRLevel interface definition, and the AMR class which oversees pre-run initialization, run execution and periodic file IO during the run. In order to use CHOMBO we must provide an implementations of the AMRLevel class. The following sections contain a look at how the AMR class operates, where the definition of AMRLevel comes in and how numerical methods have been implemented.

Despite all of the included functionality, the CHOMBO package it is by no means complete. In fact this project involved providing a number of extensions to the CHOMBO infrastructure. To implement CT numerics within the CHOMBO infrastructure, Divergence preserving restriction and prolongation operators were added for use with a face constant magnetic field. Edge constant data types were obtained and a constant interpolation operator was implemented for these. Finally, the magnetic flux conserving coarse–fine boundary conditions developed above were intergrated with CHOMBO. Due to the importance of the order of application of the magnetic flux conserving coarse–fine boundary conditions developed above, the CHOMBO code was examined in detail in an attempt to document the order in which particular pieces of the algorithm were executed. This examination lead to the discovery of a number of bugs which have in turn lead to a number of modifications to our local copy of the CHOMBO code base.

4.3 The AMR Grid Generation and Data Initialization

For a time dependent hyperbolic system the initial conditions must be provided prior to the run. The process can be thought of as a boot strapping procedure starting only with the base level, that will repeatedly apply data initialization, tagging, and regridding procedures on the current hierarchy until the maximum number of levels is achieved or the tagging procedure returns an empty set of tags. Once the bootstrapping process is finished, the resulting hierarchy is traversed, initialized, and an initial time step size is calculated at each level. This procedure allows for initial conditions which are the solution of a system

```
1     AMR::setupForNewRun()
2     make level 0 grid
3     set finest level to 0
4     for level m = 0 to min(finest level, max level):
5        for level i = 0 to m:
6           AMRLevel::initialGrid(grids i)
7        for level i = 0 to m:
8           AMRLevel::initialData()
9        for level i = 0 to m:
10          current tags i = AMRLevel::tagCellsInit()
11          union current tags i with old tags i
12       finest level = BRMeshRefine::regrid(new grids,tags,0 to m, grids)
13       if finest level > m:
14          set grids to new grids
15       for level i = 0 to m:
16          set old tags i to tags i
17    for level m = 0 to finest level:
18       current dt = new dt = AMRLevel::computeInitialDt()
```

Table 4.1: The CHOMBO grid definition and initialization procedure.

of some auxiliary differential equations, such as computing a velocity field from a vorticity field using the elliptical solver included with CHOMBO. The details are shown in Table 4.1. In this table and those which follow an italic font is used to show where we are expected to provide code. Specifically, we must provide:

AMRLevel::initialGrid(): Given a collection of boxes we must distribute them to the available processes, allocate resources such as arrays to hold the solution and initialize interlevel communication objects.

AMRLevel::initialData(): Initialize the solution arrays according to a problem specific initial condition.

AMRLevel::tagCellsInit(): Identify computational cells which should initially be refined.

AMRLevel::computeInitialDt(): Estimate the largest finite difference time step that can be taken, with out violating the CFL condition.

```
1       AMR::run()
2       while current time<end time and steps taken < max steps
3          for all levels:
4             set time to current time
5          if need to write check point:
6             AMR::writeCheckpointFile()
7          if need to write plot file:
8             AMR::writePlotFile()
9          AMR::timeStep(level 0, steps left 0, time boundary T)
10         limit time step growth,AMR::assignDt()
11         increment steps
12         increment current time by last dt
```

Table 4.2: The CHOMBO run procedure.

When providing this functionality one must keep in mind that it is important to refine critical regions right from the start of the run. Error estimates from the solution of differential equations are not available so initial tagging criteria typically are problem specific. Also, at start up the time step size can be grossly overestimated for initial conditions containing large gradients (Toro, 1997, p. 209). Thus a problem specific safety factor is often applied to the time step computed with the CFL condition for the first few finite difference steps taken. A feature of CHOMBO is that the growth of the time step is limited to some run time specified factor (with a default value of 1.1). In the case that the initial time step returned is drastically smaller than what the CFL condition predicts, a slow retreat from the initial time step up to the CFL predicted time step allows for steep gradients to smooth during a start up period avoiding numerical instability.

4.4 The AMR Run

After the initial hierarchy of levels has been generated and initialized, the time stepping process begins and repeats itself until the desired end time is reached. The CHOMBO run process is depicted in Table 4.2. The procedure illustrated is deceptively simple because recursion is used to advance the entire grid hierarchy with a single call to AMR::timeStep().

```
1     AMR::timeStep(level m, steps left, time boundary)
2     if need to regrid:
3        AMR::regrid(m)
4     if not on time boundary:
5        if subcycling needed:
6           set up subcycling
7     AMRLevel::advance()
8     AMRLevel::computeDt()
9     save current dt
10    set new dt
11    if not level max level:
12       increment steps since regrid
13    if not finest level:
14       set time boundary T
15       while steps left
16          decrement steps left
17          steps left = timeStep(level m+1,steps left, time boundary)
18          if time boundary:  set time boundary F
19    AMRLevel::postTimeStep()
20    return steps left
```

Table 4.3: CHOMBO time step procedure.

4.5 The AMR Time Step Procedure

At the heart of the AMR algorithm is the AMR::timeStep() procedure. Table 4.3 list the algorithm implemented in CHOMBO. Note that AMR::timeStep() is a recursive procedure which will advance the solution on the entire hierarchy of levels through the time increment spanned by the Level 0 step before returning to the caller. On the refined levels at least r steps are required to bring that level in sync with its parent, where r can be different at each level. However this relies on the use of a constant time step size in between synchronization points, which is not always possible while satisfying the CFL condition. If the CFL condition dictates a smaller time step than is needed to bring the current level in sync with its parent in r steps, then subcycling is needed. Subcycling becomes important when conditions are expected to change drastically on scales of the order of the step size. One example might be a chemically reacting flow. I disabled subcycling and instead will use a constant time step on refined levels relying on a conservative CFL number to insulate us from CFL condition

```
1       AMR::run()
2       while current time<end time and steps taken < max steps
3          for all levels:
4             set time to current time
5          if need to write check point:
6             AMR::writeCheckpointFile()
7          if need to write plot file:
8             AMR::writePlotFile()
9          AMR::timeStep(level 0, steps left 0, time boundary T)
10         increment steps
11         increment current time by last dt
```

Table 4.4: Modified CHOMBO run step procedure.

violations. Rather than use the time step size estimated using the level 0 solution as the base time step size, the most restrictive time step is found across all of the levels such that the CFL condition is satisfied on all levels when a fixed time step is used.

From line 8 in Table 4.3 we can see that the CFL condition is applied and the time step size is computed after the finite difference step is taken, however in between this calculation and the next advance, the solution will potentially be modified twice. First in AMRLevel::PostTimeStep() where a restriction from a finer level occurs. Second, in the next call to AMR::timeStep() if regridding occurs the solution will be potentially modified. Both of these modifications, strictly speaking, invalidate the time step size estimate. In the worst case this could lead to a CFL condition violation. The other issue here is that given this order of operations derived quantities, such as pressure, must be calculated twice, for instance they are needed in AMRLevel::computeDt(), and they also needed during AMRLevel::advance(), the potential modification of the solution in between means that they must be computed again. All of which makes this particular ordering a poor choice. Additionally, due to the importance of the order of operations when applying the magnetic flux conserving coarse–fine boundary conditions I have made changes to CHOMBO to remedy these issues. Technically speaking the time step size must not be computed until such time that the solution will not change between the time that the calculation is made and the next call

```
1     AMR::timeStep(level m, steps left, time boundary)
2     if need to regrid:
3        AMR::regrid(m)
4     AMRLevel::computeDt()
5     save current dt
6     set new dt
7     if level 0:
8        AMR::assignDt()
9     f not on time boundary:
10       if subcycling needed:
12          set up subcycling
13    AMRLevel::advance()
14    if not level max level:
15       increment steps since regrid
16    if not finest level:
17       set time boundary T
18       while steps left
19          decrement steps left
20          steps left = timeStep(level m+1,steps left, time boundary)
21          if time boundary:  set time boundary F
22    AMRLevel::postTimeStep()
23    return steps left
```

Table 4.5: Modified CHOMBO time step procedure.

to AMRLevel::advance() where the finite-difference step is actually taken. Given that the solution will not change between the time step size computation and the finite difference operation, it is a convenient location to compute the derived quantities. Both the coarse–fine boundary conditions and physical boundary conditions need to be applied prior to the calculation of the derived quantities. Thus boundary conditions are also applied here. The changes I have made to CHOMBO are shown in bold in Tables 4.4 and 4.5.

We must provide AMRLevel::advance(), AMRLevel::postTimeStep(), AMRLevel::computeDt(). These methods provide the following functionality:

AMRLevel::computeDt(): Calculate the largest time step size that can be safely taken during the next advance.

AMRLevel::advance(): Finite difference the solution by a pre-computed time step.

```
1       AMR::regrid(level m)
2       for i = m to min(finest level,max level-1)
3          AMRLevel::tagCells(tags[i])
4          save old grids
5          save old problem domains
6       new finest level = BRMeshRefine::regrid(new grids,tags,0 to m, grids)
7       if new finest level > finest level:
8          set time in new finest level
9       for i = m+1 to new finest level:
10         AMRLevel::preRegrid(new grids[i])
11      for i = m+1 to new finest level:
12         AMRLevel::regrid(new grids[i])
13      for i = new finest level+1 to max level:
14         AMRLevel::regrid(empty grid)
15      for i = new finest level down to m
16         AMRLevel::postRegrid(m)
```

Table 4.6: The CHOMBO regrid procedure.

Accumulate differences in fluxes between this level and the next coarser one according to equation (3.2).

AMRLevel::postTimeStep(): Inject solution from the next finer level and apply the correction of equation (3.3).

We have chosen a second order Runge-Kutta time stepping scheme (RK2) using hybrid Rusanov-Zalesak numerical fluxes. The Level 0 implementation is simplified compared to that of the finer levels because there is no coarse–fine boundary to deal with on level 0. Therefor I have split the implementation into two cases, one for Level 0 and the other for refined Levels. During the time stepping procedure it is critical to maintain the magnetic flux conserving boundary conditions which I have introduced above. As the description of the specific details which I have used is rather long, involved and not well suited to elegant prose, I have moved it to the Appendices 5.7 and 5.7.

4.6 The Regridding Procedure

Regridding is the phase of the time step where the grids are allowed to evolve. It may not be necessary to regrid at every time step, therefore each level may have its own regridding interval. When it is time to regrid a particular level, refinement tags are generated on that level and each finer level in the hierarchy, and a new hierarchy of grids is produced from them. It is important to remember that during a regrid the computational grid for the level being regrid will not change. Changes will only occur in the finer levels of the hierarchy. During this process we must initialize new data structures to hold the solution on the newly defined computational grid, create objects that will handle the coarse–fine interlevel communications, transfer the solution from the current grid hierarchy to the new, and initialize newly introduced computational cells. In the case that a level disappears we should free its computational resources as they are no longer needed, and may be used elsewhere. Additionally, if we are running a parallelized code it is our responsibility to distribute the work to the available computational units. CHOMBO's regridding process is shown in Table 4.6. We must provide the following functionality:

1. **AMRLevel::tagCells():** Identify cells which the next finer level should cover after refinement.

2. **AMRLevel::regrid():** Allocate data structures to hold the solution on a newly refined computational grid and initialize those data structures, while maintaining the existing solution.

The specific details regarding my implementation have been moved the Appendix 5.7.

CHAPTER 5

RESULTS AND CONCLUSION

5.1 Chapter Abstract

In this chapter I present my code in action. I have gathered four simple initial conditions which isolate individual aspects of the underlying physics of the MHD system. In each case I have obtained an analytic result with which to compare any numerically obtained solution. The cases are run without adaptivemesh refinement. This presentation documents the correctness of my implementation of the solver numerics and forms a basis for a regression suite to be utilized in the future development of the code to help ensure that code correctness is maintained and give a baseline with which to compare new numerical modifications. The last of the test cases is run with the adaptive meshing on and demonstrates that my mew extension of constrained transport numerics for use on block structured adaptive grids works. I will discuss the results of these runs and introduce future directions which I would like to explore going forward. The AMR code that I have described here is a work in progress, while much has been accomplished, there are still a number of issues which I am working on at this time. Currently, single level functionality is complete, allowing the numerics implemented to be fully investigated. Additionally AMR with a single level of refinement is operational, allowing for preliminary validation of the magnetic flux conserving boundary conditions. Here I will present a series of MHD problems which demonstrate the current functionality of the code and identify some of the strengths and weaknesses of the underlying numerical methods which have been employed.

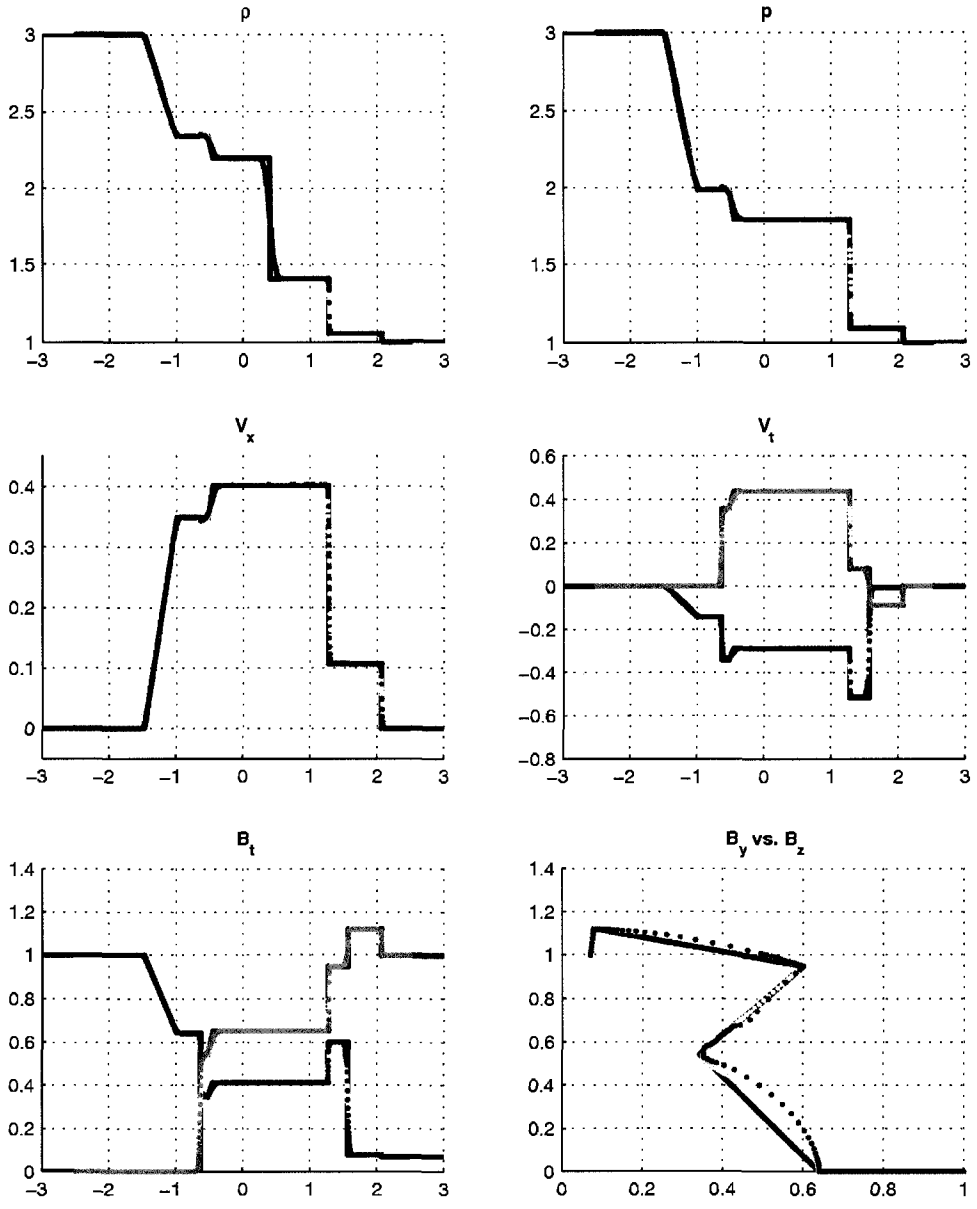


Figure 5-1: The numerical solution to the Torrilhon Riemann problem. Exact solution shown in black solid line. Red and Green dots show hybrid flux solution while yellow dots show Rusanov flux solution.

5.2 The Torrilhon Riemann Problem

The first single level validation test case is the MHD shock tube of Torrilhon (Torrilhon, 2002). We run the quasi 1-D problem as a long tube of 2000x3x3 computational cells extending from -1 to 1 along the coordinate axis. Our left and right states are as follows:

$$(\rho, \mathbf{v}, p, \mathbf{B}) = \begin{cases} (3, 0, 0, 0, 3, 1.5, 1, 0) & , x < 0 \\ (1, 0, 0, 0, 1, 1.5, cos(\alpha), sin(\alpha)) & , x \geq 0 \end{cases} \tag{5.1}$$

Where the angle $\alpha = 85.9437^o$ is the rotation of the transverse magnetic field across the initial discontinuity. These initial conditions are robust in the sense that they excite 7 MHD waves in the solution, namely slow and fast magnetosonic shock, fast and slow rotational waves, and a contact discontinuity. We set the gas gamma to $\gamma = 5/3$, the CFL number to 0.8 and evolve the solution until an end time of $t = 1$ is reached. The results are shown in Figure 5-1. The exact solution is plotted using a solid black line, while the numerical solution computed using only the Rusanov fluxes is plotted with yellow dots and the solution computed using the hybrid fluxes is plotted with red dots. In the plots of the transverse components of $\mathbf{v}$ and $\mathbf{B}$, red dots are used for the y-component computed using hybrid fluxes while green dots are used for the z-component computed using hybrid fluxes. The numerical solution is in good agreement with the exact solution. However, these plots reveal a weakness of the numerics employed. Specifically, the large smoothing of the contact discontinuity(density field at $x \approx 0.5$) and the trailing edge of the rotational discontinuities are both a concern.

5.3 The Brio and Wu Riemann Problem

A second single level validation test case is the MHD shock tube of Brio and Wu (Brio and Wu, 1988). This MHD Riemann problem is a popular code validation problem. Again, we run the quasi 1-D problem as a long tube of 2000x3x3 cells extending from -1 to 1 along

the coordinate axis. Our left and right states are as follows:

$$(\rho, \mathbf{v}, p, \mathbf{B}) = \begin{cases} (1, 0, 0, 0, 1, 0.75, 1, 0) & , x < 0 \\ (0.125, 0, 0, 0, 0.1, 0.75, -1, 0) & , x \geq 0 \end{cases} \tag{5.2}$$

These initial conditions excite a left running fast refraction fan, followed by a left running compound wave consisting of a slow shock attached to a slow rarefaction, followed by a right running contact discontinuity, followed by a right running slow shock, followed by a right running fast rarefaction fan. The compound wave is of particular interest, as its sharp narrow peak and short width should prove challenging for many finite-difference methods. We set the gas gamma to $\gamma = 2.0$, the CFL number to 0.8 and evolve the solution until an end time of $t = 0.1$ is reached. The results are shown in Figure 5-1. The exact solution is plotted using a solid black line, while the solution computed using only the Rusanov fluxes is plotted with yellow dots and the solution computed using the hybrid fluxes is plotted with red dots. This test further illustrates weaknesses of the numerics employed. Again, we see the large smearing of the contact. We also see a good deal of spurious oscillations in the region of the compound wave. It is not really clear what the source of these are. At first I thought that they might be a consequence of the way $\mathbf{j}$ is being computed. For instance one could interpret equation (2.57) as a centered three point formula, which would be unstable in the presence of discontinuities (Laney, 1998, p. 197). However, some experimentation confirmed that this was not the root cause of the instability. A best guess as to the source of the spurious oscillations at this point, is that they may have something to do with the upwind calculation of the electric field only taking into account the flow speed rather than the characteristic speeds and directions, although this is just a guess and may be off target as well. As a final note, the fully conservative form of the momentum equation could be used as the CT method prevents the destructive spurious force of equation (1.20) that the semi-conservative formulation was specifically designed to avoid, however it is not clear that such a change would have much of an impact on the spurious oscillation experienced here.

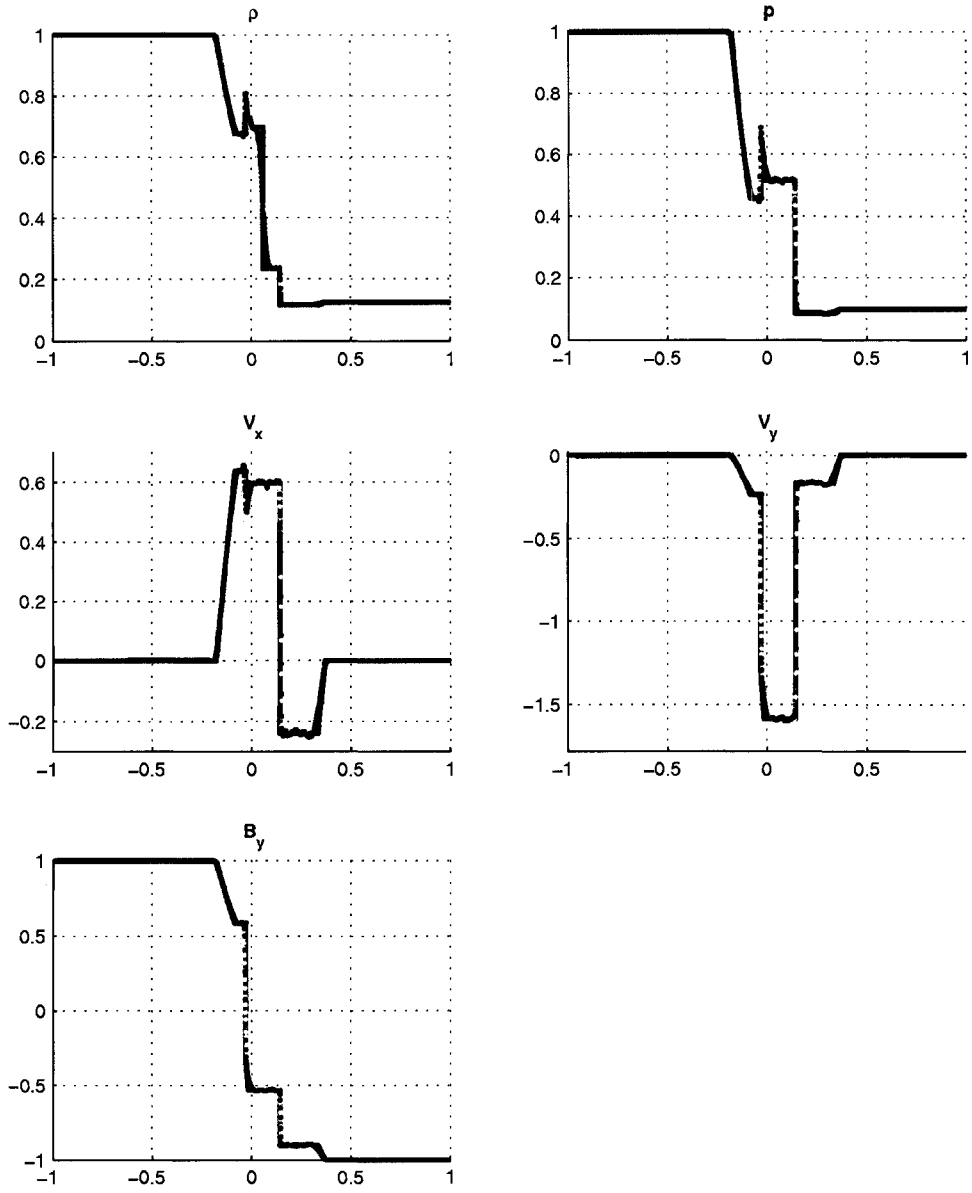


Figure 5-2: Numerical solution to the Brio and Wu Riemann problem. Exact solution shown in black solid line. Red dots show hybrid flux solution while yellow dots show Rusanov flux solution.

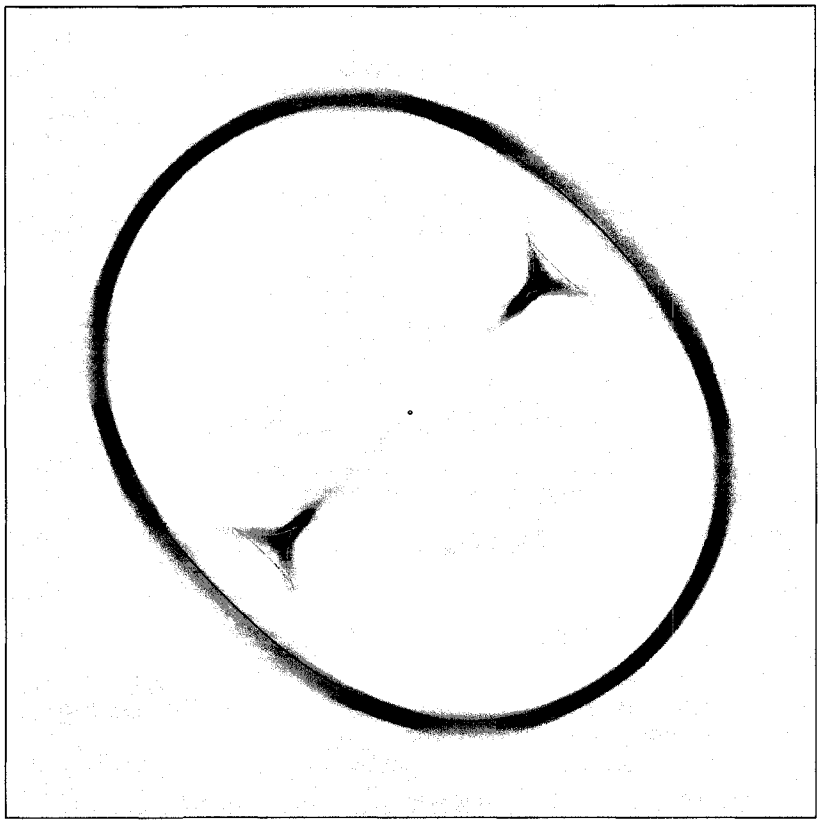


Figure 5-3: Magneto-acoustic waves. Numerical solution with Friedrichs diagram overlaid (pressure).

5.4 Friedrichs Diagrams

As described in Section 1.9, Friedrichs diagrams provide a useful single level multidimensional test case. An initial pressure disturbance launches waves into a region with uniform initial conditions. The computational grid spans from -0.5 to 0.5 in all coordinate directions with a grid spacing of $\Delta\mathbf{x} = 0.001$. The initial conditions are given by:

$$(\rho, \mathbf{v}, p, \mathbf{B}) = \left(1, 0, 0, 0, 100, 10/\sqrt{2}, 10/\sqrt{2}, 0\right) \tag{5.3}$$

The pressure is increased to $p = 100.01$ on a sphere of radius $r = 0.001$ located at the origin. We set the gas gamma to $\gamma = 5/3$, the CFL number to 0.8 and evolve the solution until an end time of $t = 0.025$ is reached. The reference solution is computed using equations (1.63) and (1.64). The results are shown in Figure 5-3. The pressure field is shown with values ranging from x (black) to x (white). The characteristic wavefronts have been plotted on

top of the result for reference. The fast mode characteristic wavefront is shown in blue, the slow mode characteristic is shown in teal, the Alfven characteristic (located at the head of the slow mode disturbances) is shown in yellow, and the initial disturbance is shown in red at the center of the figure. There is good agreement between the numerical and analytic solution.

5.5 Wave Propagation Across A Density Interface

A second single level multi-dimensional wave propagation test where the wavefront locations can be easily computed is constructed by establishing a jump in density across a y-z plane passing through the origin. The conditions on either side of the plane are:

$$(\rho, \mathbf{v}, p, \mathbf{B}) = \begin{cases} (1,0,0,0,0.1,0,10,0) & ,x < 0 \\ (0.04,0,0,0,0.1,0,10,0) & ,x \geq 0 \end{cases} \tag{5.4}$$

We will solve this problem on an x-y-plane given by $-0.5 \leq x \leq 0.5$ and $-0.5 \leq y \leq 0.5$. A velocity disturbance of $v_x = 0.01$ is introduced on a small cylinder of radius $r = 0.005$, located at $x_0 = -0.0625$. This small disturbance will excite the characteristic MHD waves. The two plots near the bottom of Figure 5-4 show the characteristic wavefronts and surfaces of normal velocity for these conditions. The fast characteristic speed is nearly isotropic, and coincident with the Alfven speed. Because the fast mode characteristic speed is isotropic the evolved wavefront will retain its initial cylindrical shape as it propagates. The change in density is set up so that the Alfven speed in the left state is $v_a = 1$ while in the right state it is set up to be $v_a = 2$. As the traveling fast wave encounters and passes through the density interface it will in part be reflected and in part be refracted according to Snell's law and Fermat's principle. I will not go into the details of how the locations of the evolved wavefront are analytically found. The reader can find the details in Levshin and kaufman (2000a) chapter 8, and (Levshin and kaufman, 2000b) chapter 1. Under these initial conditions the slow speed is much smaller than the Alfven speed and thus the slow characterstic wavefronts will not be visible over the time spanned by the numerical simulation. The results of the numerical simulation are shown in Figure 5-4 with the analytically computed

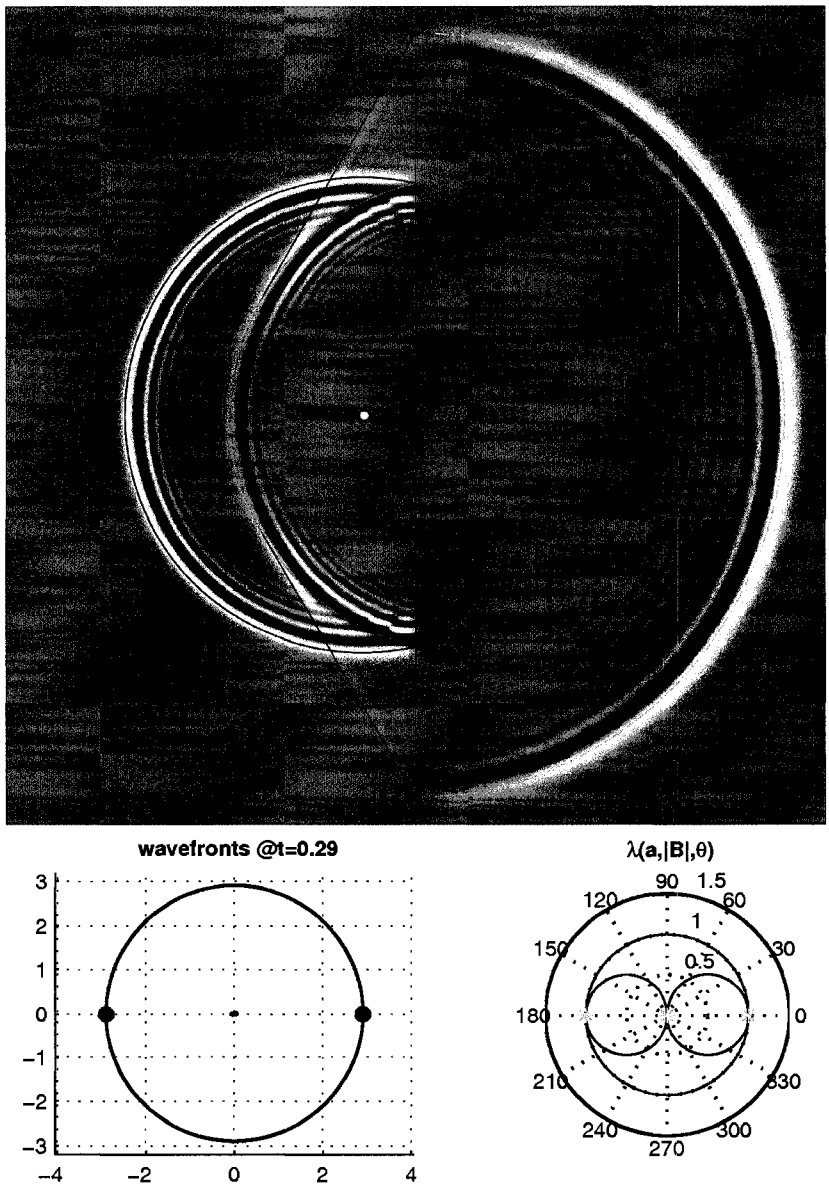


Figure 5-4: MHD wave propagation across a density interface (Top). Friedrichs diagram shown for the same conditions without the density interface (Bottom left). Surfaces of normal velocity. (Bottom right). See the legend in Figure 1-3.

wavefront locations overlaid. The x-component of the velocity field is shown and ranges from $-5.87E-6$ (black) to $3.75E-6$ (white), with the direct wavefront (blue), reflected wavefront (teal), lateral wavefront (red) and refracted wavefront (yellow) overlaid. The plots show that a good agreement is attained.

5.6 The MHD Blast Wave With AMR

We present a cylindrical MHD blast problem as a preliminary validation of the AMR features of the code. The MHD blast wave has been previously used as a validation on multi-dimensional MHD codes in Gardiner and Stone (2005). In our solution the coarsest grid is composed of 100x100x3 cells, and spans -0.5 to 0.5 in the x-y plane. The initial conditions are given by:

$$(\rho, \mathbf{v}, p, \mathbf{B}) = \begin{cases} 1, 0, 0, 0, 1, 10/\sqrt{2}, 10/\sqrt{2}, 0 & , r \leq 0.1 \\ 1, 0, 0, 0, 10, 10/\sqrt{2}, 10/\sqrt{2}, 0 & , r > 0.1 \end{cases} \tag{5.5}$$

We run the code until the fast magnetosonic wave hits the boundary at $t = 0.038$. The results are shown in Figure 5-5. We show from left to right, top to bottom: 1) the refined grids, 2) $|\mathbf{B}|$, 3) $|\nabla \cdot \mathbf{B}|$, and 4) the growth of $\nabla \cdot \mathbf{B}$. We have tagged cells for refinement where, $0.09 \leq |\mathbf{v}| \leq 0.12$, and as a result the grids closely track the wavefronts propagating into the constant state. During this run we regrid at each step so that we are more likely to highlight any inconsistencies in our method. None were found. In the plot of $\nabla \cdot \mathbf{B}$ the refined region has a noticeably larger divergence than the unrefined region. This is expected as, compared to the unrefined region, twice the number of time steps have been taken on the refined region due to the tighter CFL restriction there. We note that $\nabla \cdot \mathbf{B} \approx 0$ including at the coarse–fine interface, which illustrates that we have indeed conserved magnetic flux across the interface. Every ten steps we have sampled the minimum and maximum value of the divergence of the magnetic field and plotted the magnitude of these values (blue squares and red circles, respectively) as a time series along with a least squares fit for each(dashed and solid line, respectively). We overlay a plot of the order of magnitude prediction (teal solid line) given in equation (2.54). The slopes of the least squares lines show that growth

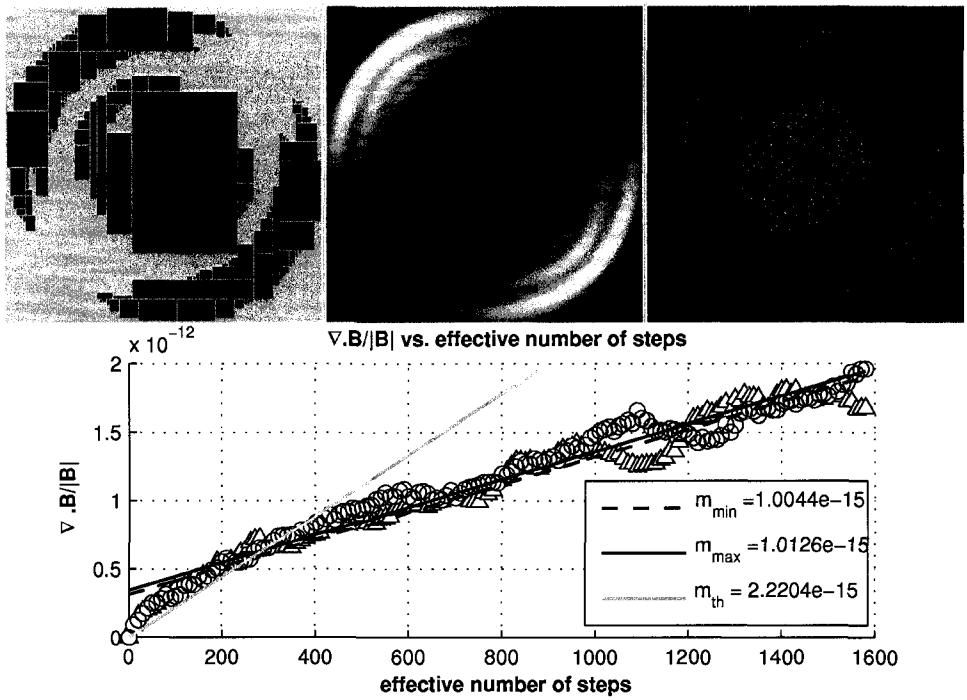


Figure 5-5: MHD blast wave with AMR. From left to right, on the top row: 1) Refined grid in black; 2) $|\mathbf{B}|$ varying from 9.64(dark) to 10.3 (light); and 3) $\nabla \cdot \mathbf{B}$ varying from -1.72E-11 (black) to 2.01E-11 (white). On the bottom row: 1) circles track $|\max \nabla \cdot \mathbf{B}|$ and triangles track $|\min \nabla \cdot \mathbf{B}|$, where both have been scaled by $|\mathbf{B}|$ and sampled at 10 step intervals; 2) Linear least squares fit(black lines) obtained using these data; and 3) light line shows the estimate given in equation(2.54).

of $\nabla \cdot \mathbf{B}$ is indeed of the order of magnitude predicted by equation (2.54). The successful completion of this test illustrates that the various pieces of the AMR code function correctly both individually and collectively as a system. However, convergence studies have yet to be completed.

5.7 Conclusion

In this thesis I have described the work done on a new AMR code for ideal MHD. I have reviewed the underlying physics of the ideal MHD system which are necessary to interpret the results of a MHD code and which make it possible to construct a few useful exact solutions for the purposes of code validation. This review included a discusion of the propagation of small amplitude waves in the MHD system as well as a thorough discussion

of MHD shocks, contacts and rarefactions and how they can be pieced to gether to obtain a solution to the MHD Riemann problem. I have reviewed some basic concepts regarding shock capturing numerical methods and numerical nonlinear stability while at the same time introducing the numerics that have been used in the new code. I have reviewed the various issues associated with the divergence free constraint in MHD codes and discussed the popular methods for dealing with the problems that occur when it is not satisfied. I have discussed in detail the constrained transport method for limitting the growth of the divergence of the magnetic flux density, and constructed an estimate of an upper bound on the growth rate of the divergence. I have discussed the fundamental concepts of block structured adaptive mesh refinement, including conservation principles across grid resolution changes. I have introduced the CHOMBO C++ framework for AMR and shown how they work and how my code makes use of them. I have introduced a new method of enforcing magnetic flux conservation across AMR coarse–fine grid interfaces, and demonstrated that it works. Finally I have shown a number of test cases illustrating the current functionality of the code, and the strengths and weaknesses of the underlying numerics. Future work will involve improving the nonlinear stability of the numerics, performing convergence studies where the new magnetic flux conservation boundary conditions are used, and porting the code to an MPI cluster environment.

BIBLIOGRAPHY

D. Balsara. Divergence Free Adaptive Mesh Refinement for Magnetohydrodynamics. *Journal of Computational Physics*, 174:614–648, 2001.

D. S. Balsara and D. Spicer. Maintaining Pressure Positivity in Magnetohydrodynamic Simulations. *Journal of Computational Physics*, 148:133–148, January 1999.

J. Bazer and O. Fleischamn. Propagation of Weak Hydromagnetic Discontinuities. *Physics of Fluids*, 2:366–378, 1959.

J. Bell, M. Berger, J. Saltzman, and M. Welcome. Three-Dimensional Adaptive Mesh Refinement for Hyperbolic Conservation Laws. *Journal of Scientific Computing*, 15(1): 127–138, January 1994.

M. J. Berger. On conservation at grid interfaces. *SIAM Journal on Numerical Analysis*, 24 (5):967–984, 1987.

M. J. Berger and P. Colella. Local adaptive mesh refinement for shock hydrodynamics. *Journal of Computational Physics*, 82:64–84, May 1989.

M. J. Berger and J. Oliger. Adaptive Mesh Refinement for Hyperbolic Partial Differential Equations. *Journal of Computational Physics*, 53:484–+, March 1984.

T. Boyd and J. Sanderson. *The Physics of Plasmas.* Cambridge University Press, 2003.

J.U. Brackbill and D.C. Barnes. The Effect of nonzero $\nabla \cdot \mathbf{B}$ on the Numerical Solution of the Magnetohydrodynamic Equations. *Journal of Computational Physics*, 1(35):426–430, 1989.

M. Brio and C. C. Wu. An upwind differencing scheme for the equations of ideal magneto-hydrodynamics. *Journal of Computational Physics*, 75:400–422, April 1988.

R. Burden and D. Faires. *Numerical Analysis.* Brooks Cole, 2001.

P.A. Davidson. *An introduction to magnetohydrodynamics.* Cambridge University Press, 2001.

P. J. Dellar. A Note on Magnetic Monopoles and the One-Dimensional MHD Riemann Problem. *Journal of Computational Physics*, 172:392–398, September 2001.

C. Evans and J. Hawley. Simulation of Magnetohydrodynamic Flows: A Constrained Transport Method. *The Astrophysical Journal*, 332:659–667, 1988.

T. Gardiner and J. Stone. An Unsplit Godunov Method for Ideal MHD via Constrained Transport. *Journal of Computational Physics*, 205(2):509–539, May 2005.

P. Godlewski, E.and Raviart. *Numerical Approximation of Hyperbolic Systems of Conservation laws.* Springer, 1996.

D. Goldberg. What Every Computer Scientist Should Know About Floating-Point Arithmetic. *ACM Computing Surveys*, 23(1):5–48, March 1991.

J.B. Goodman and R.J. LeVeque. On the Accuracy of Stable Schemes for 2-D Scalar Conservation Laws. *Mathematics of Computation*, 45:15, 1985.

David Handscom. Report no. 91/5: Interpolation and Differentiation of Multivariate Functions and Interpolation of Divergence-free Vector Fields using Surface Splines. Technical report, Oxford University Computing Laboratory, Numerical Analysis Group, 1991.

A. Harten. High Resolution Schemes for Hyperbolic Conservation Laws. *Journal of Computational Physics*, 49:357–+, March 1983.

T.Y. Hou. Why Non-conservative Schemes Converge tothe Wrong Solutions. *Mathematics of Compuation*, 62:497–530, 1994.

A. Jeffrey. *Magnetohydrodynamics.* Oliver & Boyd, 1966.

A. Jeffrey and T. Taniuti. *Non-linear Wave Propagation.* Academic Press, 1964.

J Kim, D. Ryu, T. Jones, and S. Hong. A Multidimensional Code for Isothermal Magnetohydrodynamic Flows in Astrophysics. *The Astrophysical Journal*, 514:506–519, 1999.

C. B. Laney. *Computational Gasdynamics.* Cambridge University Press, 1998.

P. Lax. Introduction to "Highresolution Schemes for hyperbolic Conservation Laws". *Journal of Computational Physics*, 135:259, 1997.

R. J. LeVeque. *Numerical Methods for Conservation Laws.* Birkhauser, 1992.

A.L. Levshin and A.A. kaufman. *Acoustic and Elastic Wave Fields in Geophysics Part I.* Elsevier, 2000a.

A.L. Levshin and A.A. kaufman. *Acoustic and Elastic Wave Fields in Geophysics Part II.* Elsevier, 2000b.

X.-D. Liu and P. D. Lax. Positive schemes for solving multi-dimensional hyperbolic systems of conservation laws. *Journal of Computational Fluid Dynamics*, 5(2):428–440, May 1996.

F. Mackay, R. Marchand, and K. Kabin. Divergence-free Magnetic Field Interpolation and Charged Particle Trajectory Integration. *Journal of Geophysical Research (Space Physics)*, 111:6205–+, June 2006. doi: 10.1029/2005JA011382.

R. S. Myong and P. L. Roe. Shock Waves and Rarefaction Waves in Magnetohydrodynamics. Part 1. A model system. *Journal of Plasma Physics*, 58:485–519, October 1997a.

R. S. Myong and P. L. Roe. Shock Waves and Rarefaction Waves in Magnetohydrodynamics. Part 2. The MHD system. *Journal of Plasma Physics*, 58:521–552, October 1997b.

K.G. Powell. An Approximate Riemann Solver for Magnetohydrodynamics (That Works in More Than One Dimension). Technical report, ICASE Semi-annumal Report No. NAS1-19480, 1994.

J. Raeder. Global Magnetohydrodynamics – A Tutorial. In J. Büchner, C. T. Dum, and M. Scholer, editors, *Space Plasma Simulation.* Springer Verlag, Berlin Heidelberg New York, 2003.

V. Rusanov. The Calculation of the Interaction of Non-Stationary Shock Waves and Obstacles. *USSR Computational Mathematics and Mathematical Physics*, 2:304–318, 1962.

D. Ryu, J. P. Ostriker, H. Kang, and R. Cen. A Cosmological Hydrodynamic Code Based on the Total Variation Diminishing Scheme. *The Astrophysical Journal*, 414:1–19, September 1993. doi: 10.1086/173051.

G. Sod. A Survey of Several Finite Difference Methods for Systems of Non-Linear Hyperbolic Conservation Laws. *Journal of Computational Physics*, 27:1–31, 1978.

E. Tadmor. Numerical Viscosity and the Entropy Condition for Conservative Difference Schemes. *Mathematics of Computation*, 43:369–381, 1984.

E.F. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics.* Springer, 1997.

M. Torrilhon. Exact Solver and Uniquness Conditions for Riemann Problems of Ideal Magnetohydrodynamics, Research Report No. 2002-06. Technical report, Seminar for Applied Mathematics, ETH Zurich, 2002.

M. Torrilhon. Non-uniform Convergence of Finite Volume Schemes for Riemann Problems of Ideal Magnetohydrodynamics. *Journal of Computational Physics*, 192:73–94, November 2003. doi: 10.1016/S0021-9991(03)00347-4.

G. Toth. The $\nabla \cdot \mathbf{B} = 0$ Constraint in Shock-Capturing Magnetohydrodynamic Codes. *Journal of Computational Physics*, 161:605–652, 2000.

G. Toth and P. Roe. Divergence- and Curl-Preserving Prolongation and Restriction Formulas. *Journal of Computational Physics*, 180:736–750, 2002.

G.B. Witham. *Linear and Nonlinear Waves.* John Wiley & Sons, 1974.

C.C. Wu. Magnetohydrodynamic Riemann Problem and the Structure of the Magentic Reconnection Layer. *Journal of Jeophysical Research*, 100(A4):5579–5598, April 1995.

S. Zalesak. Very High Order and Pseudospectral Flux-Corrected Transport Algorithms for Conservation Laws. *Advances in Computer Methods for Partial Differential Equations*, 4:126–134, 1981.

S. Zalesak. A Physical Interpretation of the Richtmyer Two-Step Lax-Wendroff Scheme and its Generalization to Higher Order. *Advances in Computer Methods for Partial Differential Equations*, 5:491–496, 1984.

APPENDICES

APPENDIX A

GLOSSARY OF AMR TERMINOLOGY

For reference the definitions that follow have been illustrated in Figures A-1 and A-2. This first of these is more useful when thinking about the finite-difference procedure, while the second is more useful for thinking about the regridding procedure.

Cell-wise intersection: Cells intersect each other cell-wise when one cell contains entirely the other. The cell-wise intersection of two cells which share a single face, a single edge, or a single corner is empty.

Face-wise intersection: Faces intersect each other face-wise when one face contains entirely the other. The face-wise intersection of two cells which share a single edge, or a single corner is empty.

Edge-wise intersection: Edges intersect each other edge-wise when one edge contains entirely the other. The edge-wise intersection of two cells which share a single corner is empty.

Box-wise intersection: Intersection is performed on the cells enclosed by the box, either cell-wise, face-wise or edge-wise.

Periodic intersection: The intersection is made after shifting the box into its periodic location. For a 3-dimensional problem periodic in all three directions, there are 26 periodic locations for each box. Think Rubik's cube.

layout: A collection of disjoint boxes.

shape: Used to describe the dimensions of individual arrays in a data container. The shape is specified by the layout boxes.

computation boxes: Layout boxes which have been expanded to include ghost cells.

computational cells: Cells in the union of all of the computation boxes.

problem domain: The smallest box containing all of the level 0 boxes given in units of a particular level.

interior cells: Computational cells which are in the cell-wise intersection with the layout boxes.

exterior cells: Computational cells which are not in the cell-wise intersection with the problem domain.

ghost cells: Computational cells which intersect more than one layout box, or intersect none of the layout boxes. The intersection is taken non-periodic cell-wise.

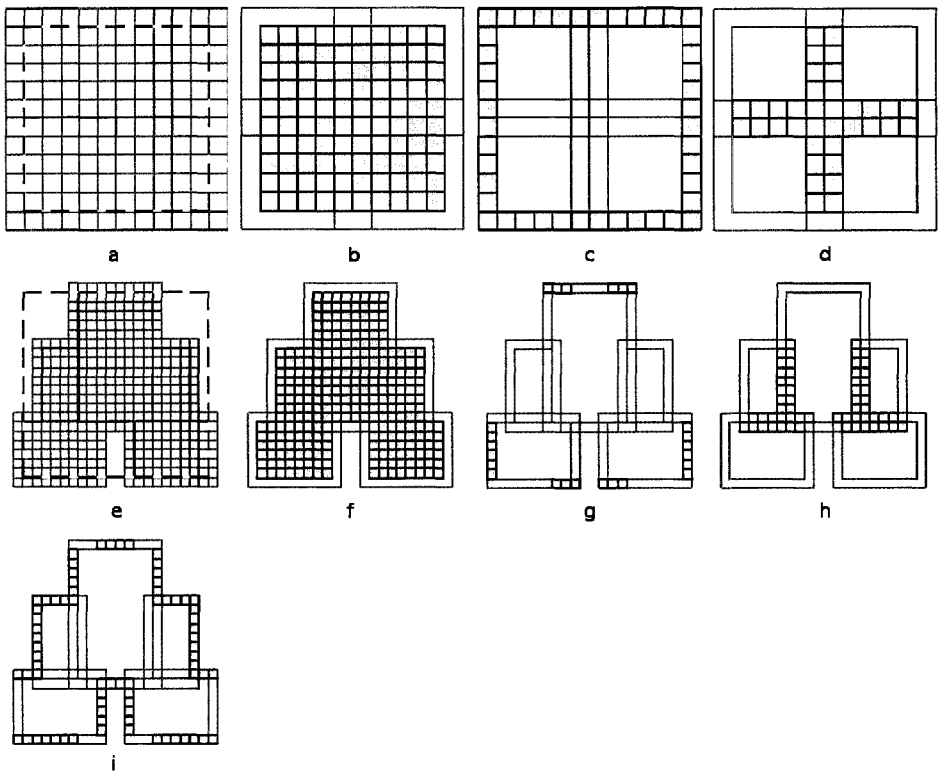
interior ghost cells: Computational cells which intersect more than one layout box.

periodic ghost cells: Computational cells which are in the periodic intersection of the layout boxes.

coarse–fine interface(CFI) cells: Computational cells which intersect none of the layout boxes and are not in the periodic intersection of the layout boxes.

exchange: Operation which copies data from the interior of each of the computational boxes into the intersection of its interior with each other computational boxes and into the periodic intersection of its interior with the other computational boxes.

flux register: CHOMBO object which stores the coarse flux and sum of the fine fluxes and computes their difference during the application of a correction finite difference step.



a) Level 0 composed of 4 boxes(black) and problem domain(dashed).

b) Level 0 interior cells.

c) Level 0 boundary condition ghost cells.

d) Level 0 interior ghost cells.

e) Level 1 composed of 5 boxes(black) and problem domain(dashed).

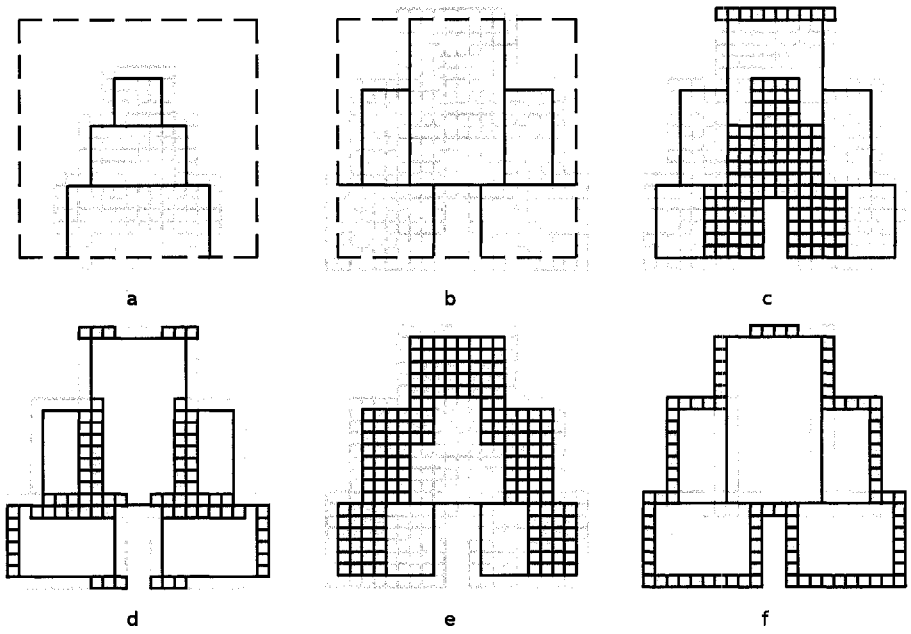
f) Level 1 interior cells.

g) Level 1 periodic ghost cells.

h) Level 1 interior ghost cells.

i) Level 1 coarse–fine interface.

Figure A-1: Groups of computational cells important in AMR. Level 0 has 4 boxes (a-d) and a single level of refinement has 5 boxes(e-i).



a) Current grid layout (black), cells (light gray), and problem domain(dashed).

b) New grid layout (black), cells (light gray), and problem domain(dashed).

c) Cells of the new layout which are in the periodic intersection of the interior of the current layout and the new layout. These are filled by a copy.

d) Ghost cells of the new layout which will be valid after an exchange.

e) Interior cells of the new layout which must be initialized via an interpolation from the next coarser level.

f) Coarse-fine interface cells which need to be filled via an interpolation from the next coarser level.

Figure A-2: Groups of cells important during the regrid operation (c-f) taking grid a to grid b.

APPENDIX B

THE AMR RK2 CYCLE ON LEVEL 0

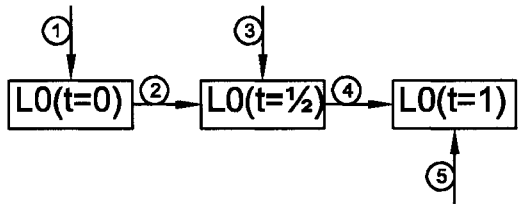


Figure B-1: Level 0 RK2 procedure.

The level 0 procedure is schematically depicted in Figure B-1. Boxes indicate data containers. Arrows indicate operations on the data. Down arrows indicate application of the boundary conditions, left arrows indicate a finite difference operation, and upward arrows indicate a transfer of data from a finer level. The solution at t=0 is given. The solution at t=1 is computed. The steps taken in my implementation are as follows:

AMRLevel::computeDt()

1. (a) If not using periodic boundary conditions:
   i. Apply boundary conditions to the cells of $\rho, \rho\mathbf{v}, U, \mathbf{B}$ which are outside of the problem domain.
   (b) Compute derived quantities $P, \mathbf{v}, \mathbf{B}_c, \mathbf{j}$ from $\rho, \rho\mathbf{v}, U, \mathbf{B}$.
   (c) Compute stable Δt on this Level.[1].
   (d) If a finer level exists:

AMRLevel::advance()

2. (a) Calculate $\mathbf{E}$ on edges enclosing interior cells from $\mathbf{B}$, $\mathbf{v}$, $\frac{1}{2}\Delta t$
   (b) Exchange on $\mathbf{E}$ fills interior ghost cells and periodic ghost cells.
   (c) Copy $\rho, \rho\mathbf{v}, U, \mathbf{B}$ into a work array $\rho^*, \rho\mathbf{v}^*, U^*, \mathbf{B}^*$
   (d) Compute predictor flux from $\rho^*, \rho\mathbf{v}^*, U^*, \mathbf{B}^*$ and $P, \mathbf{v}, \mathbf{B}_c, \mathbf{j}$
   (e) Finite difference $\rho^*, \rho\mathbf{v}^*, U^*, \mathbf{B}^*$ by $\frac{1}{2}\Delta t$ using predictor flux.
   (f) Exchange on $\rho^*, \rho\mathbf{v}^*, U^*, \mathbf{B}^*$ updates interior and periodic ghost cells.
3. (a) If not using periodic boundary conditions:

[1] Δt used for step must be the most restrictive across all levels.

i. Apply boundary conditions to $\rho*, \rho\mathbf{v}*, U*, \mathbf{B}*$ exterior cells, and faces enclosing exterior cells.

(b) Compute derived quantities $P, \mathbf{v}, \mathbf{B}_c, \mathbf{j}$ from $\rho^*, \rho\mathbf{v}^*, U^*, \mathbf{B}^*$

(c) Calculate $\mathbf{E}$ on edges of the interior cells from $\mathbf{B}^*$, $\mathbf{v}$, Δt.

(d) Exchange on $\mathbf{E}$ updates interior and periodic ghost cells.

4. (a) Compute corrector flux from $\rho^*, \rho\mathbf{v}^*, U^*, \mathbf{B}^*$ and $P, \mathbf{v}, \mathbf{B}_c, \mathbf{j}$.

(b) If a finer level exists:

i. Clear flux register.

ii. Store corrector flux.

(c) Finite difference $\rho, \rho\mathbf{v}, U, \mathbf{B}$ by Δt using corrector flux.

(d) Exchange on $\rho, \rho\mathbf{v}, U, \mathbf{B}$ fills interior ghost cells and periodic ghost cells.

(e) Initiate next finer level RK2 cycle, and wait for its completion.

AMRLevel::postTimeStep()

5. (a) If a finer level exists:

i. Create work array $\rho^\#, \rho\mathbf{v}^\#, U^\#, \mathbf{B}^\#$ with same shape as the next finer level.

ii. Conservative average from the next finer level into $\rho^\#, \rho\mathbf{v}^\#, U^\#, \mathbf{B}^\#$.

iii. Copy from $\rho^\#, \rho\mathbf{v}^\#, U^\#, \mathbf{B}^\#$ into $\rho, \rho\mathbf{v}, U, \mathbf{B}$.

iv. Reflux $\rho, \rho\mathbf{v}, U, \mathbf{B}$.

v. Exchange on $\rho, \rho\mathbf{v}, U, \mathbf{B}$ updates interior and periodic ghost cells.

APPENDIX C

THE AMR RK2 CYCLE ON REFINED LEVELS

On refined levels the RK2 cycle occurs in two procedural applications for a refinement ratio of 2. For clarity the steps of both applications are explicitly depicted in Figure C-1. Boxes indicate data containers. The letter C indicates that a data container holds solution on the coarse layout, while a letter F indicates that a data container holds solution on the fine layout. Arrows indicate operations on the data. Down arrows indicate application of the boundary conditions, solid horizontal arrows indicate a finite difference operation, dashed horizontal arrows indicate a time interpolation, and upward arrows indicate a transfer of data from a finer level. The solution at t=0, and t=1 is given on the coarse layout. The solution at t=0 is given on the fine layout. The solution on the fine layout at t=1 is computed. The procedure is as follows:

AMRLevel::computeDt()

1. (a) If not using periodic boundary conditions:
   i. Apply boundary conditions to $\rho, \rho\mathbf{v}, U$ exterior cells[2].
   (b) Apply coarse–fine interface boundary condition on CFI cells.
   If $t_0 = 0$:
   i Interpolate $\mathbf{E}$ into edges of the CFI cells.
   ii Divergence Free interpolate $\mathbf{B}$ into the faces of the CFI cells.
   If $t_0 = 1/2$:
   i Time interpolate coarse solution at t_0 into a coarsened work array $\rho^{\#}, \rho\mathbf{v}^{\#}, U^{\#}, \mathbf{B}^{\#}$.
   ii Interpolate into the CFI cells of $\rho^*, \rho\mathbf{v}^*, U^*$ from $\rho^{\#}, \rho\mathbf{v}^{\#}, U^{\#}$.
   iii Divergence Free interpolate from $\mathbf{B}^{\#}$ into the faces of the CFI cells of $\mathbf{B}$ not in the face-wise intersection of the union of the interior cells and the periodic ghost cells.
   (c) Compute derived quantities $P, \mathbf{v}, \mathbf{B}_c, \mathbf{j}$ from $\rho, \rho\mathbf{v}, U, \mathbf{B}$.
   (d) Compute stable Δt on this Level.[3].

AMRLevel::advance()

---

[2]Applying boundary condition to $\mathbf{B}$ would be redundant since the coarse–fine boundary condition needs to be applied here as well.

[3]Δt used for step must be the most restrictive across all levels.

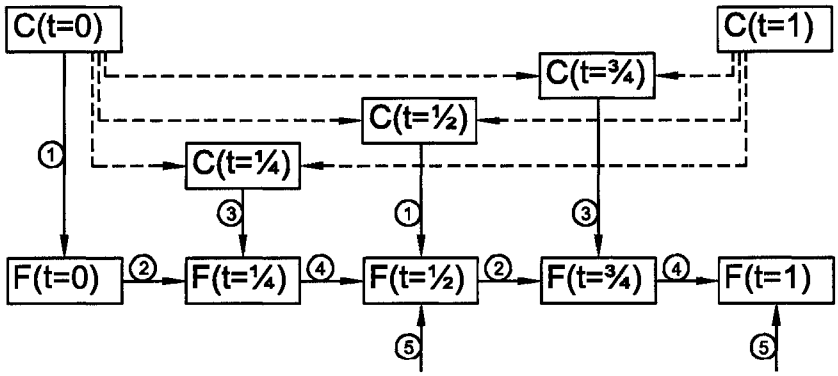


Figure C-1: Refined level RK2 cycle.

2. (a) Calculate $\mathbf{E}$ on edges of the interior cells not in the edge-wise intersection with the CFI cells, from $\mathbf{B}$, $\mathbf{v}$, $\frac{1}{2}\Delta t$.
   (b) Exchange on $\mathbf{E}$ fills interior ghost cells and periodic ghost cells.
   (c) Copy $\rho, \rho\mathbf{v}, U, \mathbf{B}$ into a work array $\rho^*, \rho\mathbf{v}^*, U^*, \mathbf{B}^*$.
   (d) Compute predictor flux from $\rho^*, \rho\mathbf{v}^*, U^*, \mathbf{B}^*$ and $P, \mathbf{v}, \mathbf{B}_c, \mathbf{j}$.
   (e) Finite difference $\rho^*, \rho\mathbf{v}^*, U^*, \mathbf{B}^*$ by $\frac{1}{2}\Delta t$ using predictor flux.
   (f) Exchange on $\rho^*, \rho\mathbf{v}^*, U^*, \mathbf{B}^*$ updates interior ghost cells and periodic ghost cells.
3. (a) If not using periodic boundary conditions:
      i. Apply boundary conditions to $\rho, \rho\mathbf{v}, U$ exterior cells[4].
   (b) Apply coarse–fine interface boundary condition:
      i. Time interpolate coarse solution at $t = t_0 + \frac{1}{4}$ into a coarsened work array $\rho^\#, \rho\mathbf{v}^\#, U^\#, \mathbf{B}^\#$.
      ii. Interpolate into the CFI cells of $\rho^*, \rho\mathbf{v}^*, U^*$ from $\rho^\#, \rho\mathbf{v}^\#, U^\#$.
      iii. Divergence-free interpolate from $\mathbf{B}^\#$ faces into $\mathbf{B}$ faces of the CFI cells not in the face-wise intersection of the union of the interior cells and the periodic ghost cells.
   (c) Compute derived quantities $P, \mathbf{v}, \mathbf{B}_c, \mathbf{j}$ from $\rho^*, \rho\mathbf{v}^*, U^*, \mathbf{B}^*$.
   (d) Calculate $\mathbf{E}$ on edges of the interior cells not in the edge-wise intersection with the CFI cells, from $\mathbf{B}$, $\mathbf{v}$, Δt.
   (e) Exchange on $\mathbf{E}$ updates interior ghost cells and periodic ghost cells.
4. (a) Compute corrector flux from $\rho^*, \rho\mathbf{v}^*, U^*, \mathbf{B}^*$ and $P, \mathbf{v}, \mathbf{B}_c, \mathbf{j}$.
   (b) Finite difference $\rho, \rho\mathbf{v}, U, \mathbf{B}$ by Δt using corrector flux.
   (c) Increment the coarser level's flux register using the corrector flux.
   (d) If a finer level exists:

[4]Applying boundary condition to $\mathbf{B}$ would be redundant since the coarse–fine boundary condition needs to be applied here as well.

i. Clear flux register.
ii. Store corrector flux.

(e) Exchange on $\rho, \rho\mathbf{v}, U, \mathbf{B}$ updates interior ghost cells and periodic ghost cells.

AMRLevel::postTimeStep()

5. (a) If a finer level exists:
   i. Create work array $\rho^{\#}, \rho\mathbf{v}^{\#}, U^{\#}, \mathbf{B}^{\#}$ with same shape as the next finer level.
   ii. Conservative average from the next finer level into $\rho^{\#}, \rho\mathbf{v}^{\#}, U^{\#}, \mathbf{B}^{\#}$.
   iii. Copy from $\rho^{\#}, \rho\mathbf{v}^{\#}, U^{\#}, \mathbf{B}^{\#}$ into $\rho, \rho\mathbf{v}, U, \mathbf{B}$.
   iv. Reflux $\rho, \rho\mathbf{v}, U, \mathbf{B}$.
   v. Exchange on $\rho, \rho\mathbf{v}, U, \mathbf{B}$ updates interior and periodic ghost cells.

APPENDIX D

THE REGRIDDING PROCEDURE USED

Details of our regridding implementation are as follows:
AMRLevel::regrid():

1. If new layout is empty:
   (a) Free existing resources(dynamically allocated memory, inter-level communication objects etc...).
   (b) Stop here.
2. Make a work array with the same shape as the current layout with no ghost cells.
3. Copy current $\rho, \rho\mathbf{v}, U, \mathbf{B}$ into the work array.
4. Update the shape of the current data structures and inter-level communication objects to match that of the new layout.
5. Interpolate $\rho, \rho\mathbf{v}, U$ from next coarser level into interior and ghost cells.
6. Interpolate $\rho, \rho\mathbf{v}, U$ from next coarser level into CFI cells.
7. Copy existing $\rho, \rho\mathbf{v}, U$ from work array.
8. Exchange on $\rho, \rho\mathbf{v}, U$.
9. Compute overlap mask array.
   (a) Create overlap mask array, with ghost cells.
   (b) Initialize the overlap mask to zero.
   (c) For each computational box in the overlap mask:
      i. Intersect with each layout box.
      ii. Set overlap mask to one on the intersection.
10. Copy existing $\mathbf{B}$ from work array.
11. Divergence-free interpolate interior cells of $\mathbf{B}$ according to overlap mask.
12. Exchange on $\mathbf{B}$ fills interior and periodic ghost cells.
13. Divergence-free interpolate the CFI cells of $\mathbf{B}$[5].

[5]This need not be done as the first step of the RK2 cycle is to fill these cells.